Oasys Ltd Software Suite

Version 12.1

Update and Release Notes

Oasys Ltd

The Software house of Arup

Contents

		Page
1	**Introduction**	**1**
1.1	Compatibility with LS-DYNA releases	1
1.2	Hardware platforms supported	1
1.3	Graphics hardware supported	1
1.4	FLEXlm Licensing	1
2	**Bugs Fixed**	**3**
2.1	PRIMER	3
2.2	D3PLOT	69
2.3	T/HIS	93
2.4	REPORTER	100
2.5	Shell	103
3	**Enhancements**	**104**
3.1	PRIMER	104
3.2	D3PLOT	128
3.3	T/HIS	133
3.4	REPORTER	140

1 Introduction

The Oasys Ltd LS-DYNA Environment 12.1 suite, dated May 2015 is a full release which supersedes all previous releases.

This document lists the enhancements that have been made during development, and the bugs that have been fixed (since 11.1). Each "case id" referred to in these lists is unique and should be cited if more information or clarification are required.

1.1 Compatibility with LS-DYNA releases

The 12.1 release has been used and tested with LS-DYNA version R7.1, and supports all the keywords in volumes I, II and III of the R7.1 User's Manual dated May 2014

LS-DYNA R7.1 is now the default keyword output format from PRIMER.

1.2 Hardware platforms supported

The complete Oasys Ltd LS-DYNA Environment 12.1 is available on the following platforms:

Windows	Vista, Windows 7, Windows 8/8.1	32 bit
	Vista 64, Windows 7 64bit, Windows 8/8.1 64bit	32 and 64 bit
Linux	Redhat Enterprise 4.x Redhat Enterprise 5.x Redhat Enterprise 6.x SUSE Enterprise 11.0	32 and 64 bit

For simplicity of support and maintenance only the RedHat Enterprise 5.x build has been released on the website as this has been found to run on most machines.

If you require a different Linux build, please contact us.

1.3 Graphics hardware supported

The default graphics protocol used on all platforms in the table above is OpenGL, which gives good quality 3D performance on a wide range of hardware.

"Legacy" X-Windows (2D only) graphics drivers remain in PRIMER and D3PLOT on Linux platforms since these are used by a small minority of clients who display over a network of mixed machine types.

1.4 FLEXlm Licensing

The version 12.1 software has been compiled using version 11.11.0 of FLEXlm (the same as the version 11.0 and 11.1 releases)

The version 11.0 licence file and licence daemons are backwards compatible with both the 10.x and 9.x software releases. This means that any existing 10.x and 9.x software will continue to work with the new licence files and servers.

Note :

1) On Windows the version of "lmgrd.exe" that is installed with the version 12 software been changed to 11.11.1. Version 11.11.1 is supported on Windows Server 2012 while the 11.11.0 release shipped with the version 11 software isn't.

2) We recommend that you update all of your Oasys Ltd LS-DYNA environment license servers with the files shipped with the version 12.1 software.

2 Bugs Fixed

The enhancements and bug-fixes for each program have been broken down into a series of topics. Within each topic enhancements and bug-fixes are listed by case ID (most recent first).

2.1 PRIMER

2.1.1 Bugs Fixed in 12.0

Airbag

- **Case 25804**

It was possible to corrupt airbag reference geometry data when animating *BOUNDARY_PRESCRIBED_MOTION within PRIMER AND displaying the reference geometry. This has been corrected.

- **Case 24836**

When reading a model containing MAT34 definitions that were not used by any elements PRIMER would show a dialogue box asking the user to choose an option. This was confusing and so has been replaced with a default action and a message for each material.

- **Case 24474**

PRIMER could crash when visualising *INITIAL_AIRBAG_PARTICLE_POSITION. This has been corrected.

- **Case 24472**

PRIMER would fail to read a model containing an airbag particle card if the card referenced individual chambers (CHM_ID) but did not reference an owner chamber card (CHM).

This is an error in the model, but it is better to continue reading and issue an error rather than just failing to read. PRIMER now does this.

- **Case 23749**

PRIMER could crash when creating chambers when daisy chaining from an airbag particle card you are also creating. This has been corrected.

- **Case 22264**

The *DEFINE_CPM_CHAMBERS individual chambers edit panel was restrictive in how to add part sets. This has been corrected.

Assembly

- **Case 24458**

Reading a part tree assembly and the JavaScript model methods Import() and ImportInclude() overwrote the model filename and path. This has now been fixed.

Attached

- **Case 23513**

The Attached.SetEntity() JavaScript function did not work for contacts (i.e. you could not turn on/off the 'find through' flag for contacts in JS). This has been corrected.

Also the Attached.SetEntity() function could reset all switches to OFF when called for the first time. This has been corrected.

Belts

- **Case 24533**

The seatbelt Auto-Refit process got a bit confused if asked to refit an existing belt that:

- was made from a mixture of 1d belt elements and shells.

- did not have a retractor at its start on the B-Post

It would tend to treat the initial stretch from B-Post anchorage to shoulder slipring as having mixed shell / 1d belt elements, rather than using pure 1d belt elements.

This was because the 'work out what is in existing belt' algorithms did not consider this arrangement, and got a bit confused.

Now fixed: end points that are not rettractors but nevertheless use all 1d belt elements to the next slipring are now treated in the same way during auto refit.

- **Case 24515**

The (seatbelt) New Dummy capability in V11 did not work very well, was difficult to understand, and was of only marginal usefulness.

It has been completely rewritten in V12 and is much improved:

- The panel layout has been revised to make functionality clearer, and in particular better visual feedback of the current status and what still needs to be done has been much improved.

- Various internal bugs have been fixed, making the whole process much more robust.

This capability should now be genuinely useful, and should greatly simplify the problem of replacing a dummy in a belted situation.

- **Case 24512**

An example model in which the labels of beams and 1d seatbelt elements overlap was provided. Prior to output PRIMER detects this and warns that Dyna's default of putting beams on springs and 1d seatbelt elements, using the same labels, means that this clash is not allowed and offers to renumber things.

If this renumbering option is used then 1d belt elements are renumbered correctly to avoid a clash, but the and seatbelt elements in the belt fitter *BELT_MESH cards now no longer encompass all 1d belt elements in that seatbelt definition. This doesn't matter until the deck is written out, reread and the belt refitted since the write out / reread cycle recalculates which 1d belt elements are in the belt definition from the first/last labels, and this is no longer correct. The

refit fails to find enough label space in its label 'slot' because some of the 1d belt elements no longer 'belong' to it.

The problem arises because of the vulnerability of using [first] and [last] labels to describe a range of 1d belt elements, which is an expedient solution to the problem that there is no *SET_SEATBELT keyword in LS-DYNA. This implicitly assumes that the belt elements will occupy a contiguous range of labels within those limits and, crucially, also assumes that 1d belt element labels will not be moved from inside that range to outside it. In this instance some were relabelled to lie outside that range, leading to their being 'lost' by the belt fitter following the keyword deck write/reread sequence.

This problem has been much reduced, if not totally solved, by adding a scan for what the first and last labels of a seatbelt definition actually are immediately prior to writing it out to the keyword file. This means that any relabelling that may have taken place will be accounted for, and the first / last labels will be truly representative of what is in the belt.

It is not a total solution because a sufficiently perverse relabelling exercise could result in some elements not originally in the belt being accidentally now included. However this is fairly unlikely as the automatic relabelling built into PRIMER will attempt to minimise label changes, and it would require a fairly perverse user intent on major changes to trigger this problem.

This logic is applied to all definitions under the post-END *BELT cards that use first/last syntax: 1d seatbelt elements, sliprings, retractors, database cross-sections and node sets used for nodal rigid bodies.

- **Case 23947**

*ELEMENT_SEATBELT_SLIPRING is normally bi-directional, with the belt able to pull through in either direction. However it is possible to set a DIRECT flag to limit this to 1->2, or 2->1.

In the case of a 2D slipring using a non-zero DIRECT value it seems that the location of EDGSET (on *SECTION_SHELL) for that stretch of 2d belt elements influences the values reported for pull-through at the slipring, and to give the correct results EDGSET should be located on the 'downstream' (pulled through to) side.

Unfortunately this can conflict with the requirement to locate EDGSET at a retractor if that is in this section of belt, so this is probably a bug in LS_DYNA and has been reported to LSTC. Pending resolution of this PRIMER adopts the following logic:

- If this stretch of 2d belt elements contains a retractor then EDGSET is located there.

- Otherwise if the stretch contains a slipring EDGSET is located at the downstream side as determined by DIRECT.

This logic may need to change when LSTC fix this bug, but the above is correct for PRIMER 12 development as of Feb 2014.

- **Case 23352**

The EDGSET definition on the *SECTION_SHELL card determines the layout of a stretch of 2d belt elements, and if this set is empty or refers to nodes / elements that are not on a 2d belt LS-DYNA terminates with an error in the input phase.

If the seatbelt fitter is used to remesh an existing 2d belt into a different type, ie 1d belt elements and/or shells, it will leave the unused *SET_SHELL definition behind, although it should

remove any EDGSET definition on this card. It has been reported that EDGSET gets left behind in this situation, although it has not proved possible to replicate this problem.

However an extra function has been added to the checking of *SET_SHELL that will detect an 'orphan' EDGSET definition that does not refer to 2d belt elements, giving the option of removing this.

- **Case 21957**

There is a limit to the labels that may be used on *ELEMENT_SEATBELT_SLIPRING definitions due to formatting limitations in LS-DYNA.

For a 1D slipring:

Neither node nor seatbelt elements may have a label > 99,999,999

For a 2D slipring:

Node set may not have a label < -9,999,999

Shells sets may not have labels < -99,999,999

Previously PRIMER did not trap these limits, but they have now been added to the standard

- **Case 20073**

Two ergonomic problems related to seatbelt fitting have been fixed:

(1) In the 'Fit' panel you can make changes to the belt path in a mini editor, but these changes were 'scratch' and not saved in the belt definition unless you did an 'Accept'. Since you had to 'Save' in the editor this was a bit counter-intuitive, and a user could reasonably expect that the 'save' would be enough to make his changes permanent.

Therefore the internal logic has changed so that this is now the case: changes to the belt path become permanent as soon as you 'Save' them.

(2) When moving belt points by dragging it is sometimes useful to swap temporarily to 'Quick pick' mode (eg using short cut key 'q') in order to blank things that are in the way.

This works, but it moves the 'ownership' of the cursor picking process from the belt fitting window to the main graphics one, and in order to revert to dragging points it is necessary to tell the code that you want to go back to this mode. Clicking on the [+] symbol in the top left corner of the seatbelt fitting box, or repeating the 'q' short cut key, would achieve this; however some users also assumed that by clicking again on the appropriate belt path editor button such as 'modify coords' this too would return picking ownership to the seatbelt panel. The latter didn't work because the editor thought 'I'm still in modify coordinates mode', and therefore that no action was required to re-acquire ownership of cursor picking.

This has been fixed by making sure that clicking on any of the mouse-driven belt editor buttons (Delete points, Modify Coords, Control Twist) will always re-acquire ownership of picking if necessary. (This change went into V11.1 as well as V12)

Blanking

- **Case 24660**

IGES Curves might fail to be selectable for screen-picking, either explicitly or by area selection, and this might appear to occur in a rather random fashion.

This was caused by internal errors when marking these curves as 'visible' when drawn, and has been fixed.

Case

- **Case 26154**

If a model contained more than approximately 50 child cases then PRIMER could generate errors during keyword input, and not process the deck correctly.

The error was due to populating an internal matrix of case combinations with NxN (where N = number of child cases) case combination entries, even though the vast majority of these would never actually occur. This overflowed internal storage space and caused the error.

This has been fixed by only creating internal combinations of 2 or more child cases, where a combination is a nested CASE_BEGIN/END definition, when they actually occur.

Checking

- **Case 25724**

Custom checks did not work for materials if there were different material types in the model (which is very likely). Now fixed.

- **Case 25676**

Model check may incorrectly report error that MAT169 solids do not align with shell planes when THKDIR setting is zero. Now fixed.

- **Case 25050**

Problems have arisen with the joint coincidence check in LS-DYNA which requires that pairs of nodes on joint types 1 - 7 are within 1.0e-3 of each other. This check is not related to the size or units of the model, and is an absolute value.

PRIMER checks for this and will identify any nodal pairs at joints which breach this limit, but this check has occasionally gone wrong in two ways:

(1) PRIMER has found the nodal separation to be < 1.0e-3, so acceptable, but LS-DYNA finds it to be less and thus generates an error.

(2) Passage of an input deck through PRIMER can cause subtle small changes to nodal coordinates that cause nodes to separate very slightly, once again triggering this check in LS-DYNA.

The problem seems to arise because of the single precision storage used for nodal coordinates in PRIMER, which limits the precision of numbers to just over 7 decimal places.

This is fine when a node is close to the origin, but when it is moved some distance away the magnitude of its coordinate gets larger, and the precision required to compute the 1.0e-3 absolute value becomes more doubtful.

Testing reveals that for coordinates between 8192.0 and 16384.0 the problem is critical: values below 8192.0 retain sufficient precision, and values above 16384.0 lose the 4th decimal place

entirely, however within that range the 4th decimal place is essentially noise, and this can cause errors in the calculation.

To fix this two changes have been made:

(1) The existing nodal coincidence check for joints in PRIMER examines the distance of the joint from the origin, and if it is further than 8192.0 units away it uses a test of 0.5e-3 rather than 1.0e-3.

This takes into account errors in the precision with which numbers are stored, and makes it slightly more likely that an error will be flagged and the coordinates of the nodes merged to an average value, but a difference of 0.5 in 10,000,000 will make no practical difference to a model.

(2) Prior to keyword output there is a second, silent check for joint nodal coincidence.

This considers each [x,y,z] axis of nodal pairs at joints separately, and merges the coordinates in a given axis if:

- Their magnitude is greater than 8192.0

- They are less than 1.5e-3 apart.

The intention of this check is to remove any spurious errors that may arise because of small changes in precision introduced by PRIMER, but to do so in a way that minimises changes to the model.

- **Case 24611**

An existing error check that throws a warning if some but not all nodes on any shell have a reference geometry definition has been modified. This warning will now only be issued if said shell is actually used in an airbag.

- **Case 24598**

Contact Automatic General containing only seatbelt elements, incorrectly gave message that slave side was empty. Now fixed.

- **Case 24324**

Autofix to remove empty set from *DATABASE_HISTORY_NODE,etc could cause crash. Now fixed.

- **Case 23883**

The error message issued when solid element formulation 13 is applied to a part of material type that does not support it has been clarified. In these cases, the element formulation is silently converted to type 10 which can cause modelling errors.

- **Case 23882**

When is defined for discrete beam of non-zero length, the node is now tested for alignment with N1-N2 and and error message given if this is the case.

- **Case 23452**

The autofix which enforces coincidence for joint nodes did not work well in the case of a node shared by multiple joints. Fixing one joint would break another. The logic has been upgraded to average the position of all the nodes concerned.

Additionally, an out of colinear cylindrical joint (warning) will only be fixed where the nodes are not shared with other joints.

Compare

- **Case 25422**

In Model Modified it is possible to sketch changed items from the report tree, and this was being slow for the case of modified PARTs in some models if many parts had been changed.

This was because of the way extra nodes on rigid parts were being drawn and this has been recoded to make it much faster.

The Sketch operation in this context will now look at the 'Stop' button periodically, so if a sketch is still taking a long time the user can terminate the operation by using STOP.

Connections

- **Case 26152**

Filtering the object menu for connection by connection title could result in SIGSEV. Now fixed.

- **Case 25319**

The description for constants SPOTWELD_SOLID in the JavaScript API manual has been corrected to signify solids/spotweld layer.

- **Case 25021**

When creating tied contacts for connection entities, a penalty based contact is often created in parallel to a constraint based one to ensure any nodes that do not tie due to a constraint clash are tied. PRIMER v11 created *CONTACT_SPOTWELD_OFFSET when creating a penalty contact in parallel to a *CONTACT_SPOTWELD contact, but _OFFSET cannot be applied to a SPOTWELD type contact. PRIMER will now create a *CONTACT_TIED_SHELL_EDGE_TO_SURFACE_OFFSET contact instead.

- **Case 24887**

When auto-creating connection information on top of existing spotwelds, PRIMER did not consider the panels any spotwelds were meshed into when populating layer information for the connection entity. This could lead to a connection entity with unexpected layer information, and therefore labelled as BAD. This has been corrected.

- **Case 24595**

Connections data can now be written out to IGES format file from the command line.

- **Case 24454**

The 'Ignore parts' option on the connection read panel did not ignore parts as intended. This has been corrected.

Constraints

- **Case 25497**

Creation of NRBs using screen area pick with a large number of nodes was found to be very slow. Now fixed.

Contacts

- **Case 25910**

Illegal floating value error messages no longer appear if Check > Options input boxes contain the valid or strings.

- **Case 25737**

PRIMER gave an error by mistake if creating/checking a *CONTACT_2D_AUTOMATIC_SINGLE_SURFACE card without specifying the SIDM field (not required for the single surface contact type). This has been corrected.

- **Case 25706**

Null Box Id field when set type (MSTYP/SSTYP) doesn't support Box definition.

- **Case 24664**

The MPP method switch for penetration check under check > contact > options did not work resulting in the default MPP method being applied always with possible incorrect report for SMP contacts. Now fixed.

- **Case 23718**

PRIMER now considers *ELEMENT_SHELL offsets during contact penetration checking.

Note that LS-DYNA, and hence also PRIMER, only consider offsets during the contact penetration calculation if:

Field CNTCO is set on *CONTROL_SHELL.

The contact surface is not an AUTOMATIC type.

In addition nodes on the slave sides of NODES_TO_SURFACE contacts do not 'inherit' the offset of their parent shell.

(See also case 15950)

- **Case 20622**

Error checks that determine whether nodes lie on or behind rigidwalls now consider the effect of SHLTRW in *CONTROL_CONTACT.

Contour

- **Case 25113**

Made the following changes:

When ESORT=1 on *CONTROL_SOLID, and you do a contour plot of ELFORM,

- If the ELFORM on *SECTION_SOLID is 19 and the element is a wedge (PENTA), displayed as 21

- If the ELFORM on *SECTION_SOLID is 20 and the element is a wedge (PENTA), displayed as 22

- A TETRA with ELFORM=19,20,21 or 22 is omitted and throws an warning.

- **Case 23999**

Bands on the contour ramp for all relevant CT-SI plots now respond to cursor-driven actions.

Left clicking on a particular band restricts the contour to entities corresponding to the current band. It is possible to quickly interrogate contours corresponding to different bands by left-clicking on various bands successively.

Middle clicks reset the contour.

Right clicks reveal additional contour refinement options.

- **Case 15071**

Timestep for 4 noded tet elements with formulations 16, 17 was incorrectly calculated. Now fixed.

Control

- **Case 25308**

-Sorted out non-static control cards order to be alphabetical.

-The type code for CONTROL_MPP_DECOMPOSITION_ARRANGE_PARTS was CMDAP, change it to CMDP.

- **Case 23985**

Changing the field TYPE (part or part set) to a value which is out of range was crashing the PRIMER, now TYPE is set to default(PART type) if it is out of range.

Curve menu

- **Case 21755**

When importing 'xy' data in the loadcurve editor the file selector will now show .txt files and .csv files (previously only .txt files were shown)

Cut section

- **Case 25879**

The sequence:

1. Read model
2. Define and turn on a cut section
3. Turn off cut section
4. Delete model
5. Read new model
6. Turn on cut section

Could cause the new model not to be displayed in cut section form until some explicit graphics command was issued.

This was because the management of the cached graphics for cut sections wrongly believed that the new model was up to date, whereas it needed an explicit redraw. Now fixed by better cached graphics management.

- **Case 25794**

The cut section property calculation will derive Area, centroid, Ixx, Iyy and Ixy properties for cut sections, and these are calculated correctly. However when calculating properties for a doubly symmetric equal section such as a cube or a circle, where Ixx == Iyy, problems could arise when reporting the angle between the principal axes Iuu and Ivv and the element axes Ixx and Iyy.

Clearly in a double symmetrical equal section where Ixx == Iyy then the off-diagonal term Ixy should be zero, and the principal terms Iuu and Ivv should be equal to Ixx and Iyy. This also means that the angle theta between the principal terms Iuu/vv and the element axis terms Ixx/yy is indeterminate and should be treated as zero.

In real life such sections, when made up from finite elements and nodal coordinates, tend to have small dimensional errors which mean that there is a small residual Ixy term, and also that Ixx and Iyy may be slightly different. PRIMER knows this and has a tolerance check to detect this and to clamp the angle theta to zero, but this was set too tightly and did not pick up the ill-conditioning case reliably enough.

The tolerance check has been improved so that not only does it look at the radius of the Mohr's circle, but also now also considers the relative magnitudes of Ixx/yy and Ixy.

A separate problem with the optional calculation of the torsional constant J for a thin-walled closed section made of shell elements was also detected and corrected.

- **Case 21706**

Certain beam types, notably elform = 2 and elform = 12, have their engineering properties supplied as explicit Area, Iss, Itt and Irr values, and do not have shape information.

When 'true' beam section display is used PRIMER attempts to generate a shape that gives a reasonable indication of the beam's size and orientation. This cross-section shape is also used when calculating section properties in the cut-section panel.

Historically this has been calculated as a rectangular hollow section, and this usually works, but can give poor results in some situations:

+ The properties are non-physical. For example the quoted area is not compatible with the quoted Ixx and Iyy values even when a solid rectangular section is used. (Such properties can be correct if they represent a composite section, but there is no way of knowing this.)

+ The properties can be represented as a thin-walled rectangular hollow section, but its aspect ratio becomes ridiculous because they actually refer to an I beam.

+ The torsion constant, J, is not considered when synthesising a section. Therefore if torsion constants are to be computed in the cut-section properties panel they are likely to be wrong.

Various improvements have been made:

- The synthesised cross-section will now be an I beam if the engineering properties make this shape seem more plausible than a rectangular hollow section. The switch-over point between these two section types is when the RHS height : depth ratio exceeds 2:1 or 1:2.

- Better error trapping for 'non physical' properties has been added, with reversion to the best achievable solid rectangle being used rather than a 'hollow' section in which the walls overlap.

- In the cut-section property calculation the supplied engineering properties (Area, Iss, Itt, Ist, J) are used explicitly, rather than attempting to calculate properties from the synthesised section shape.

Database

- **Case 26122**

Sometimes the Create button could be greyed out when creating *DATABASE_CROSS_SECTION cards using the 'feature line' mode. This has been corrected.

- **Case 25292**

Picking nodes for NID when there is no Title in DATABASE_HISTORY_NODE_LOCAL_ID no longer causes crashes.

- **Case 25054**

Set Defaults button for *DATABASE_CPM_SENSOR now works.

- **Case 24970**

Model check now reports correct number of Database categories when DATABASE_EXTENT cards are present.

- **Case 24935**

Editing the DCOMP field (overtyping rather than using the pop up) in *DATABASE_EXTENT_BINARY now works correctly.

- **Case 23478**

Listing for DATABASE cards has been tidied up.

Deleting

- **Case 25372**

A model containing *DEFINE_CONNECTION_PROPERTIES, in which reference is made to many materials, but where the card itself is referred to by other materials, failed to 'clean up'.

This was really a more generic problem that MATerial is an 'owner' category when it comes to selection (blank a material and you expect to see its parts and elements disappear), but a 'child' category when it comes to deletion checking. Adding some extra logic to make this distinction in the right places fixed the problem.

- **Case 23674**

Modified the part deletion logic in relation to DEF_TO_RIGID cards.

- **Case 17653**

The 'Remove, Cleanup' operation has special settings for empty PART and SET cards, the default being that if these are found they are marked for removal since they are redundant.

However it is possible to change the default behaviour to 'no action' which should leave both parts/sets and references to these unchanged in this situation. This works correctly where references to parts/sets exist, but if no such references exist then the parts/sets are still marked for deletion.

This is inconsistent behaviour and has been fixed: where *SET or *PART cards exist, but these have no contents and are also not referenced then if 'no action' is chosen then they will no longer be marked for cleanup.

Dialogue command

- **Case 25483**

When using the command line to create LS-DYNA seatsquash models you could not turn the global damping option back on after you had already turned it off. This has been corrected.

Dummies

- **Case 25952**

Historically PRIMER has marked 'stop angle' values on *CONSTRAINED_JOINT_STIFFNESS_GENERALIZED that lie outside the range +/-180 degrees as an error.

This is incorrect - or at least overly conservative. Since LS-DYNA keeps track of the number of rotations a joint undergoes during an analysis there is no need to restrict stop angles to any particular value, and in fact angles greater than 360 degrees might make sense.

However during initialisation of Joint Stiffness definitions at time zero the *initial* angle can only be evaluated in the range +/-180 degrees, so it makes sense to clamp the initial orientation of these definitions to that range. Therefore the following changes and additions have been made:

- The error message for a Joint stiffness stop angle outside the range +/-180 degrees has been removed from the joint stiffness check function.
- Input of explicit stop angles in the joint stiffness editing panel and keyword editor will now accept values outside the +/-180 deg range.
- The dummy positioner will now allow stop angles outside the +/-180 deg range to be input, but issues a warning that during positioning PRIMER will clamp movement to +/-180 deg to prevent 'wrong side of zero' errors during initialisation in LS-DYNA.
- Dummy checks now include a test for a given initial orientation being outside the initial stop angle range, and also for stop angles outside the +/-180 deg range when articulation about more than one axis is permitted.

- **Case 25837**

With Dyna Dummy Positioning and Dyna Seatsquash, if the dummy contains type 6 beams, it is possible that the resulting LS-DYNA analysis would not run in LS-DYNA, due to the type 6 beams not being deleted during the rigidification process when they could be. This has been corrected.

- **Case 23726**

When tying in values for the 'neck' or 'hip' node during 'dyna' dummy positioning it was possible to type in a node that did not exist in the model. This has been corrected.

Edit

- **Case 23736**

The *ELEMENT_BEAM edit panel will now remember settings such as keyword options and values for fields vx across multiple 'Create' instances as long as the panel isn't dismissed.

- **Case 23576**

In *BOUNDARY/*CONSTRAINED keyword if you go to any keyword panel(eg. BOUND->ACOUSTIC_COUPLING->LIST->LIST_ALL) and select LIST_ALL, it lists all the keywords which use sub-types. Now it is fixed.

Elements

- **Case 25831**

PRIMER was not calculating the correct mass for *ELEMENT_DISCRETE_SPHERE_VOLUME elements. Now fixed.

Errortrapping

- **Case 25135**

When deleting a model that contained *COMMENT keywords PRIMER did not return the memory for the comment correctly and errors occurred. Now fixed.

- **Case 24867**

A spurious error message was given if a node was edited and renumbered from the database history node panel. Now fixed.

- **Case 23774**

Crash in primer while editing CONTROL_FORMING_ONESTEP card. Now fixed.

Geometry

- **Case 25936**

PRIMER failed to read a specific IGES file due to a character string problem. Now fixed.

- **Case 24892**

Geometry tessellation has been improved. Fewer errors should now be given for surfaces.

- **Case 24534**

If geometry was oriented then cut sections using omit for both sides of the cut would not draw anything. Now fixed.

Graphics

- **Case 25005**

Graphics could not get refreshed after using the node keyword editor. Now fixed.

- **Case 24616**

Previous versions of PRIMER would not reset local element axes visibility and associated options when the shortcut key 'k' was pressed. This has now been fixed.

- **Case 22220**

When setting the 'display factor' in any Oasys Ltd software it is possible to choose a value that makes the user interface of the software unreadable.

To prevent this happening there is now a confirmation dialogue box - at the unchanged scale value - that requires you to confirm that the revised setting is to be kept. It also offers the option of reverting to the original factor.

Image

- **Case 24330**

From Windows Vista onwards the 'Aero desktop' changes the way that pixel data in images is created and stored, and one side-effect of this is that the existing 'Copy to clipboard' function available from the top left [-] popup menu on all sub-windows fails to read 3d graphics generated by OpenGL. As a result using this on a graphics window tends to leave a blank space where the picture of the model should be, although the window surrounds will be drawn correctly.

Capturing ordinary GUI windows works fine, and the various options to generate files under 'Image' also work correctly.

This has been fixed in V12 by performing the extra work necessary to capture the 3D graphics when copying window contents to the clipboard.

Include

- **Case 25943**

PRIMER could give an incorrect error message about entities being outside specified include ranges when running a model check on a model containing *ELEMENT_MASS_PART entities and containing PRIMER label ranges. This has been corrected.

- **Case 21845**

Include declash now allows the user to merge duplicate nodes (and keep the duplicate).

- **Case 21449**

Historically PRIMER has permitted the category INCLUDE FILE to be selected for deletion, and doing so will select all its contents correctly, however the include file itself will not be deleted. The same is true if MODEL is selected for deletion. An annoying consequence of this is that *DEFINE_TRANSFORMATION cards referenced from *INCLUDE_TRANSFORM can therefore not be deleted, and in addition any *PARAMETERs used on these cards will also be locked against deletion.

This is because Include files inside PRIMER are a special category that did not respond well to the normal generic processing used for deletion, meaning that they could not be deleted this way.

To fix this:

+ a new option 'Empty include file action' has been added to the Delete and Cleanup removed panels.

+ this defaults to 'No action', meaning leave an empty include file, which is no change to existing behaviour.

+ however the user can choose to 'Delete empty file', in which case the include file itself will be deleted if empty.

Include transform

- **Case 26124**

PRIMER could crash when read a model that contained *INCLUDE_TRANSFORM's which referenced *DEFINE_TRANSFORM's which had POS6P or POINT based ROTATE type transforms and the include file referenced in the transform has nested includes. This has been corrected.

- **Case 24565**

*INCLUDE_PATH definitions in INCLUDE_TRANSFORM include files did not work. Now fixed.

Installation

- **Case 23994**

PRIMER opens temporary files for various purposes, and on Windows platforms these will be called pr_tmp_nnnn where nnnn is a number in the range 1 to 9999, and will reside in the user's home directory, or in the OA_TEMP directory if that is defined.

If PRIMER crashes, or is killed in an abnormal fashion (for example by killing the controlling terminal window) then these files do not get cleaned up, resulting in a build-up of files in that area. If the limit of 9999 files is reached PRIMER will generate error messages on start up.

The workaround is simple enough: just delete the files, but this should not be necessary. Therefore the temporary filename allocator now has some logic that inspects the age of existing files pr_tmp_nnnn, and if they are older than the stipulated expiry date - by default 31 days - then the file is deleted and the name re-used.

The expiry date can be modified using the preference

oasys*temp_file_expiry: n

Where n is a number of days in the range 1 to 10,000. If n is set to zero then testing for file expiry is switched off.

Keyword

- **Case 25119**

*ELEMENT_SEATBELT_PRETENSIONER problem with PTLCID - fixed it by changing ptlcid=NULL when sbprty=2,3

- **Case 24554**

*LOAD_BODY_VECTOR is now supported by PRIMER.

- **Case 24542**

Checks added to *DATABASE_FSI, *DATABASE_FSI_SENSOR and *DATABASE_CPM_SENSOR.

- **Case 24477**

Made change so that FTYPE for *INTERFACE_SPRINGBACK can now be 0,1,2,10,11,12.

- **Case 24283**

Data for EOS_IGNITION_AND_GROWTH_OF_REACTION_IN_HE that is consistent with LSDYNA version 950 and later can now be input.

- **Case 24144**

When a part uses section point source, it is correct to define *EOS card. PRIMER incorrectly removed the reference to this card when the part edit panel was invoked. Also we incorrectly gave warning when such a part had no elements.

- **Case 23884**

Changing the field TYPE (part or part set) to a value which is out of range was crashing the PRIMER, now TYPE is set to default(PART type) if it is out of range.

- **Case 23805**

Sorted errors related to Keyword editors of *MESH_VOLUME, ICFD_BOUNDARY.

- **Case 23661**

Fixed: *CESE_MAT_GAS not read by PRIMER.

- **Case 23660**

Fixed: *CESE_PART not read by PRIMER

- **Case 23491**

Fixed it with case 22937, at least one solid element should be selected to create hex spotweld.

- **Case 23353**

Fixed: labeling issue with *ELEMENT_DIRECT_MATRIX_INPUT cards.

- **Case 23095**

PRIMER ignored large memory (e.g. 900000000) as memory allocation, and shows garbage value in *CONTROL card. Now fixed

- **Case 22296**

Fixed: PR12 fails to read *CONTROL_DYN_RELAX card.

Kwd editor

- **Case 26142**

If the keyword editor is asked top display data for cards with many rows, in this case *INITIAL_STRESS_SHELL with

- 4 integration points on plan

- 5 integration points through the thickness

- 97 extra variables per integration point

So a little under 200 values per element, then it did not set aside enough buttons to display more than 2 cards worth of data, even when using 'all on one row' format.

This limit has been raised so that at buttons for at least 40 rows are allocated

Labels

- **Case 24798**

Label ranges were not obeyed during multiple split if a shell was already outside the valid label range. Now fixed.

- **Case 24584**

When changing the include file on an edit panel while creating an entity the label of the entity could be set to zero. This has been corrected.

- **Case 24033**

In the ENTITY panel, where what is drawn is controlled, it is possible to choose what a labelled item is 'Labelled with'. This can be more than item's label, extending to its model, material id, etc.

For entity type NODE these 'labelled with' switches were not working properly, meaning that the graphics might not be updated correctly to show the current setting. This has now been fixed.

Local axes

- **Case 25006**

Local material angles for shells defined with the _MCID keyword option could be shown incorrectly in certain situations. This has been fixed.

Macro

- **Case 25146**

If a macro was run from a JavaScript and PRIMER was started with either the -batch or -exit command line arguments PRIMER would exit when the macro finished instead of when the script (that called the macro) finished. Now fixed.

- **Case 23986**

When playing a macro with Pause commands 'Button box not yet defined' error messages were occasionally shown. This has been fixed.

Mass

- **Case 25392**

The count and mass of included part/nrb _inertia displayed on the assign mass panel was incorrect if sub-group massing was in use, though the calculation was itself correct. Now fixed.

- **Case 25089**

When using the 'mass prop' feature, it was possible to get the wrong answer when selecting *ELEMENT_MASS elements. This has been corrected.

- **Case 24551**

Selection of an include file in the mass property calculator resulted in inclusion of any lumped mass on the nodes of the include, irrespective of whether or not the element from which the mass derived is in the selected include. This has been fixed so that, unless the option is set to include attached mass on elements, mass derived from elements outside the selected include will be excluded.

Materials

- **Case 25347**

PRIMER could crash if importing a material to a material edit panel, and the 'Filter' option is used. The crash would only occur if clicking on the 'Filter' button before typing in the filter string. Now fixed.

- **Case 24487**

Fixed: The field SBUCK for MAT_203 has a default and therefore zero value is now allowed.

- **Case 22284**

Fixed the units of fields & of MAT 196.

Mechanism

- **Case 24871**

Mechanisms in PRIMER provide several ways of 'driving' motion, and one of these is to define a new angle for a HINGE or LINE joint. The mechanism positioner will rotate the attached assemblies to achieve the desired angle.

In a few cases the direction of rotation went the 'wrong' way, in the sense that if clockwise rotation was requested what was actually produced was anti-clockwise motion.

This was due to an error in calculating what constitutes positive motion of a body about a vector: the existing algorithm could fail in cases where the orientation of the bodies was not well-conditioned.

The correct calculation method has been substituted and rotation should now take place reliably in the requested direction.

- **Case 24857**

(This case is related to case 24871).

There was a report of mechanism movement 'hanging', or refusing to achieve the desired angle, when using the feature that allows rotation motion of a joint to be 'driven' to a new angle. We don't have the model to test.

A very similar bug, reported in case 24871, in which driven rotation went the 'wrong way' has been corrected. The cause of this was a failure to calculate the correct rotation angle and direction for 'driven' motion, and it could produce symptoms identical to those described in this case.

Because we don't have an example model for this particular problem we cannot say with certainty that it has been fixed, but it seems very likely that it will have been, so this case is being closed.

Memory

- **Case 25991**

Testing revealed that the internal allocation of memory used a strategy that was 'too small' for very large models, meaning that when these were deleted the process of returning memory to the operating system could cause PRIMER to 'hang', or at least to take a very long time.

This problem would only show up for models with more than around 20m nodes and elements, which is beyond current production practice, so it is unlikely that it will have been encountered during normal use.

The memory allocator has been modified so that it adjusts automatically to this very large models, meaning that model deletion should now run at a reasonable speed. No user intervention is required.

- **Case 25803**

PRIMER did not return memory for *BOUNDARY_PRESCRIBED_FINAL_GEOMETRY cards correctly which meant that if a model containing lots of these cards was read and deleted many times PRIMER could use excessive memory. Now fixed.

Menus

- **Case 25349**

Running PRIMER with the command line '-batch' option (note, not the same as -d=batch) does the following:

- Starts the graphical user interface and normal graphics

- Suppresses the display of the master PRIMER window

In other words it is running normally, but you don't see the window.

For historical reasons this also turns off the 'menu auto confirm' logic which gives the default response to any 'stop and confirm' messages.

This means that anything which stops and asks the user for a response will lock up PRIMER since the window with the message, and the button you would click on to continue, are not visible.

Therefore a new command-line option '-auto_confirm' has been added which works as follows:

- If, and only if, used in conjunction with '-batch' it turns on the automatic confirmation of the default response at these 'stop and confirm' panels, meaning that execution will continue.

- In all other contexts, ie if '-batch' is not also used, it is totally ignored.

Merge

- **Case 25705**

When merging nodes if both nodes had *DATABASE_HISTORY cards then PRIMER would leave the duplicate after merging. The *DATABASE_HISTORY card on the slave node is now deleted.

- **Case 24929**

The option to collapse quads to trias did not work in node merge. This has been fixed.

- **Case 24338**

In Remove, Merge nodes there was poor ergonomic behaviour in the following circumstances:

Database contains two models, only one of which is active.

Merge nodes is in use in the active model.

The user makes the other model active in Model, List

The user found that he could not select items from the newly active model for merging, since the still open selection menu applied only to the original model. Worse, clicking on MERGE again did not reset this.

A trap had been added to ignore 'click on Merge if already merging', since to do so would restart the process losing all items currently selected for merge, and it was assumed that a second such click would be accidental.

This logic has been changed to a warning message explaining what restarting the merge process entails, and giving the option to cancel or continue the process.

In addition the status messages given in the feedback buttons have been amended to show which model is currently being processed, since this was not originally shown.

(All the above does not change the underlying functionality of Merge)

Mesh

- **Case 25718**

If a quad was split into 4 quads the local axes for the 4 shells created were not consistent. Now fixed.

- **Case 24673**

A crash has been fixed in cobweb mesh.

- **Case 24600**

For 'element detach', when the 'auto' mode was selected, detach did not work if the auto number was set to 1. This has been corrected. Auto mode also did not work for solid elements. This has also been corrected.

- **Case 22617**

In extrude shell panel, drop-down says 'no of shells', now updated to 'no of solids'.

Metal forming

- **Case 23758**

Inconsistent behavior with 'Part Selection' on Tool->FORMING menu, part selections were retained if Model is changed using text box but reset if Model is changed via Model selection panel. Now it's fixed.

Model build

- **Case 26119**

IHI style build-from-csv, when used in conjunction with the AUTO vertical option, was writing INITIAL_VELOCITY_GENERATION in a cumulative fashion. In other words, the first master model would get one card, the second would get that plus another card and so on. This has now been fixed.

- **Case 24543**

Build from csv in modes where the input is only XY data did not calculate the corresponding Z value very accurately on models where the mesh is coarse. The calculation has been improved to use the proper parametric formulation and will now give an accurately interpolated value.

- **Case 24304**

A redundant 'Options' button has been removed from the 'Save Database' panel.

Model modified

- **Case 25040**

The part compare option 'Select parts from modified model' did not work correctly. Instead of comparing just selected parts it compared all parts. This has been corrected.

- **Case 24364**

For some open-ended keywords, Model Modified failed to identify that fields have been changed. Now fixed.

- **Case 22210**

Model modified fails to report changes to expressions in *DEFINE_CURVE_FUNCTION. This has been corrected.

Nastran

- **Case 25658**

Fixed faulty logic of appending .dat at the end of include files.

- **Case 23760**

PRIMER could give warnings if multiple Nastran files were read. Now fixed.

- **Case 23689**

PRIMER was crashing when reading in a Nastran file which had something like 'TITLE =' line. Now fixed.

Nodal rb

- **Case 23998**

When creating NRBs and other constraints using 'Pick N closest nodes' or 'Pick with tolerance' PRIMER will now select the closest nodes in 3d (model) space. The previous use of 2d (screen) space could result in large constraints accidentally being created through the depth of the model.

Additionally the test for node visiblity in Hi/Shaded mode has been made rigorous.

- **Case 23579**

The PRIMER 'Cleanup' tool will no longer offer to remove 2-noded CONSTRAINED_NODAL_RIGID_BODY, _SPOTWELD, and _RIVET definitions that connect 'centre' nodes of other NRB spiders.

Node

- **Case 25776**

The node edit panel includes an option to create nodes at the centre of a hole. The logic used for this function was producing incorrect results when nodes were clustered around one side of the hole. This has now been fixed.

Orient

- **Case 25359**

Interpolation option is should not be available for Orient->translate/rotate/... with 'Copy' on - geryed out the buttons accordingly.

- **Case 25358**

Orient=>Translate (with interpolation on, and 'part only' selected) when applied to nodes, the interpolation effect is not seen on the parts attached to nodes. Now fixed.

- **Case 25093**

Fixed: Undo is greyed out in confirm Orient panels after doing project to line and trans-rot functions.

- **Case 24552**

Rotating an *INITIAL_STRAIN_SHELL card using orient could corrupt data and cause PRIMER to crash. Now fixed.

- **Case 24546**

Orient with interpolation of nodes applied to more than one model in the same orient session could result in operations on one model affecting the nodes of another. Now fixed.

- **Case 24354**

Added display of individual and cumulative movement of distance translated/rotated/scaled in orient panel.

- **Case 24353**

Sketching of selected nodes disappeared after orient, even though they were still selected. Now fixed.

- **Case 24340**

The 'Orient' menu has a 'Node Drag' button. Clicking on this button will no longer close the 'Orient' menu.

- **Case 24312**

Orienting a model with the copy option turned on did not copy contacts if the 'elems in the same part' option was used. Now fixed.

- **Case 22408**

When orient copy is applied to a part in an include with the option to make a new part, the section and material cards are automatically copied as well. The new part will be in the same include as the original but the section and material were ending up in the master file. Now they will be in same include as part.

- **Case 22339**

Orient copy with make new part option has been fixed to work for latent parts. Previously a proper installed part was required.

Parameter

- **Case 25128**

When the additional mass on a part, defined by the *ELEMENT_MASS_PART card, was defined by a parameter the calculated mass of the part was not updated if the value of the parameter was changed.

Caused by the routines that update the parameter not 'telling' the model that some cached data might have changed, resulting in the old (cached) mass value continuing to be used.

Fixed by making sure that this flag is updated whenever a parameter that is referenced has its value changed.

- **Case 25001**

When a model contains many parameters the popup list of eligible parameter names that appears when you type '&...' into a text box can get very large. Prior to V12 there was no way of managing this popup if it was larger than the enclosing window.

This popup (and others in similar contexts) will now have scroll-bars if they are larger than their parent window.

In addition it is now possible to use wild-card syntax when entering parameters into a text box. '?' replaces a a single character and '*' replaces multiple characters, and by using this syntax you can control the list of candidate parameter names displayed in the popup listing.

- **Case 24525**

The operation INCLUDE, new child could go wrong if:

+ The destination model contained parameters

+ The new child include file referenced those parameters

During the import operation the new include file is temporarily read into a scratch model prior to checking and import into the destination model, and if this scratch model does not 'know about' parameters in the destination model then errors can arise.

This has been fixed by copying all all parameters in the destination model into the scratch model prior to reading the new include file so that references to them can be resolved correctly. Special logic stops them triggering label clash checks, and also prevents their being copied back into the destination model.

- **Case 23356**

IN a PARAMETER_EXPRESSION PRIMER will accept:

either A^B

or A**B

syntax to mean 'A to the power of B', but LS-DYNA will only accept the Fortran syntax of A**B.

This is now detected as an error while checking parameters, and there is an autofix to replace '^' with '**'.

(PRIMER will continue to accept either syntax so that it remains able to read and evaluate these definitions, as rejecting them during input could cause problems.)

- **Case 16682**

When PRIMER encounters a parameterised field before the true value of the parameter is known it adopts the following policy to cope with this:

+ A value of zero is inserted as a temporary parameter value, unless the usage is in a label field.

+ In a label field a special internal 'unknown label' value is used instead.

+ When keyword input is complete and true parameter values are now known PRIMER goes back and rebuilds the internal data of the cards in question using the correct values.

This process works well so long as the use of zero does not invalidate the card format in any way, ie is illegal, or the field in question does not dictate the format of the rest of the card.

For example some *AIRBAG and *MAT cards have a variable number of rows that is dictated by earlier data fields, *PART_COMPOSITE needs to know how many layers are to follow, and so on. If the key data fields in these situations are temporarily assigned zero then the rest of the card will be read wrongly, and no amount of subnsequent rebuilding from internal data will work.

From V12 onwards there is an option under READ, OPTIONS to 'pre-scan' the input deck for parameters. This works as follows:

+ The input files are scanned and *PARAMETER cards are read, but all other cards (other than *INCLUDE to access include files) are ignored.

+ This means that all values of parameters are then 'known'.

+ Then the deck is reread properly, ignoring *PARAMETER cards since these are now known, and correct parameter values can be substituted immediately in fields where they are used.

This solves the 'used before defined' problem, but it comes at a cost: the pre-scan for parameters is faster than a normal read, but still takes a significant amount of time, which is why it is not the default behaviour. However it is the recommended solution for decks that will not otherwise read correctly.

Note that PRIMER V12 will always endeavour to write out parameters before they are used, so it is possible that reading in and writing out such a deck may solve the problem for future read operations by moving *PARAMETER keywords to earlier in the deck.

This cannot always work, for example where *PARAMETER cards are placed in an include file, and this include file is read after other include files which reference the parameters, the problem will remain since PRIMER does not automatically resequence include files.

Part replace

- **Case 24981**

The current include could be changed randomly after a part replace, and when closing the main part panel. This has been corrected.

- **Case 24036**

Operations such as part replace which create primer connections from existing welds, have been modified so the creation is only performed in the local area rather than across the whole model.

- **Case 23690**

When doing an assembly replace with a target model containing latent nodes which exist at the same label in the source model, it was found that these nodes are incorrectly merged, causing unwanted connection to an nodal rigid body. The source model is now renumbered to avoid clashes wrt latent items as well as installed ones.

- **Case 23418**

When part replace is applied to a single solid part, the process deleted any coating shells on the target part and imported any on the source part. If the source part had no coating shells this could result in unwanted shell deletion, so now no action is taken if the source part has no coating shells.

Part tree

- **Case 26086**

Changing the colour of a PRIMER assembly (for example through the part tree) did not work. This has been corrected.

- **Case 25911**

When the mouse is hovered over an item in the Part Tree that is capable of being sketched, then PRIMER will highlight it in the graphics window. (Logic similar to predictive picking.)

This works correctly, however when the mouse is moved out of the Part Tree window the highlighted item is not 'un-sketched', which can be undesirable, particularly if - as in the case of an include file -sketching has drawn many things.

This has been corrected: moving the mouse out of the Part Tree window will 'un-sketch' anything that has been highlighted in the graphics window.

- **Case 25375**

Cutting and pasting DAMPING_GLOBAL and DAMPING_MODAL in Part Tree Contents mode now works.

- **Case 25301**

Cut and paste LOAD_BODY in part tree (contents mode) now works.

Pedestrian

- **Case 26043**

The pedestrian markup script will now calculate offset lines by default.

- **Case 25967**

The pedestrian markup script could fail if the vehicle parts selected meant the upper and lower bumper lines could not be created. This has been fixed and a warning message is now written if this is the case.

- **Case 25914**

The pedestrian markup script would fail to build models if CSV file specified by the user contained spaces in the pathname. This has been fixed.

- **Case 24972**

The pedestrian markup script would crash if the output csv file used a pathname that contained directories that didn't exist. It now creates the directories so the file can be written.

- **Case 24672**

In the pedestrian markup script you can set the requested beam length for creating the lines. This was really the 'grid size' between sections through the vehicle, but if a beam was created at a large angle to the X or Y axes then the beam could get long, resulting in a loss of accuracy.

This has been improved in v12 so that beams are created with the requested beam length.

- **Case 24499**

In the pedestrian markup script, if you tried to write the impact points to a CSV file that was open in Excel the script would crash. This has been fixed.

- **Case 24496**

If the pedestrian markup script can't reach a WAD length a warning message is displayed giving possible reasons why it failed. An additional reason has been added to explain that the X axis should point backwards.

- **Case 23992**

The pedestrian markup script for the EuroNCAP grid protocol was not moving the location of grid points in the bonnet-windscreen gap as described in section 3.9.6. It now does.

Penetrations

- **Case 24876**

PRIMER ignored STF on the *PART_CONTACT card when doing penetration checking. This has been fixed.

Preferences

- **Case 23654**

In PRIMER (and other software) it is possible to override the default directory in which checkpoint files are written by using the

primer*checkpoint_dir: (also oasys*checkpoint_dir: and so on)

preference. This can also be done by setting the environment variable CHECKPOINT_DIR.

The only problem was that it didn't work! The logic to set that environment variable internally from the preference was not working correctly.

This is now fixed. However users should note that if an alternative directory is defined then it must exist, it will not be created. If it does not exist, or is read-only, then checkpoint files will continue to be written in the default location.

Properties

- **Case 25571**

PRIMER could crash on Linux when clicking on the 'Import from file' button on the properties store/save panel. This has been corrected.

Quick pick

- **Case 25723**

When using quick pick to type in entities you wish to ONLY sometimes it did not work if the entities you types in were currently blanked. This has been corrected.

Read

- **Case 25345**

Models using *INCLUDE_PATH_RELATIVE allow a relative path for include files to be added, but this pathname was not being dealt with correctly either during normal keyin or during the various 'scan' operations.

Now fixed.

- **Case 24643**

PRIMER could hang when reading an IGES file if some of the surfaces contained corrupt data. This is now fixed. Additionally offset surfaces (IGES type 140) are now supported.

- **Case 24521**

Model scan would fail if the include files contained duplicate control cards. Now fixed.

Renumber

- **Case 25351**

When selection the 'master include' to renumber using 'Renumber selection' or 'Clipboard renumber' the resulting panel which should list all entities within the master file was empty. This has been corrected.

- **Case 25078**

PRIMER has a check for parts with the same ID and NRB's, and also has an autofix to relabel the NRB's to remove the clash. The autofix gets a new label using the current include layer, whereas it should use the include layer that the NRB lives in. This has been corrected.

- **Case 25057**

Relabeling in the 'Renumber selected' panel could lead to duplicate labels in entities if changing from 'Start from' to 'Offset' after specifying the start/offset value. This has been corrected.

- **Case 24520**

A new preference and a program option have been added that can prevent renumbering of node sets used by 2D sliprings into values of eight digits or more. Such node sets are legal in more recent LS-DYNA versions (971R7.1+). Therefore, the new option only comes into effect for versions R7.0 and older.

- **Case 24142**

Keyword *ALE_MULTI-MATERIAL_LIST is unusual in LS-DYNA in that it does not have an explicit label, but each row of entries within it becomes implicit label i, where i is the order of appearance. This implicit label is then used to reference that row on other *ALE cards, *DATABASE_ALE and *SET_MULTI.

As a consequence PRIMER treats *ALE_MULTI-MATERIAL_LIST definitions as being 'labelled' because this is true. However unlike other items referred to by label these definitions require special treatment which 'protects' them from renumbering, since a model containing N such definitions must always label them 1 to N.

This special treatment was missing, with the result that a user could relabel them in some circumstances (eg Model, Renumber) and this could cause all sorts of problems, including

'hanging' PRIMER if the increment was very large. This was because, in order to preserve the 'label is order of appearance' rule, PRIMER would create the intervening definitions as latent entries.

This has been fixed by recategorising these cards internally as 'not explicitly labelled', despite the fact that they do effectively have labels that are used to refer to them.

This will protect them against conventional renumbering, however it will not protect them against accidentally typing in a large label when referring to them on other *ALE cards, *DATABASE_ALE or *SET_MULTI. Users should exercise care in this respect.

- **Case 23954**

Logic has been added so that under normal operation PRIMER will avoid giving *PART cards and *CONSTRAINED_NODAL_RIGID_BODY cards clashing labels.

- **Case 23940**

When creating section or material which uses a character string for its label in a model with include label ranges defined, the checking function incorrectly reported the items as having out of range labels. PRIMER now excludes these from the label range check.

- **Case 23069**

The include numbering table no longer warns about out-of-range database history cards and their children when the option to lock database history cards against renumbering is invoked.

Rigidify

- **Case 24303**

Rigidify could leave errors in the model if the parts to rigidify used 'chained' nodal rigid bodies or if they contained beams which had 3rd nodes in a nodal rigid body. Both these issues have been fixed.

Scripting

- **Case 26080**

Various panes or tabs in the FMH script now automatically resize to optimum dimensions

- **Case 25721**

Part.ClosestNode() would crash if used with a beam part if some of the beams had less than 3 nodes. Now fixed.

- **Case 25463**

The LCO option for *ELEMENT_DISCRETE was not supported in JavaScript. It has now been added.

- **Case 25394**

In very rare cases the debugger crashed when pressing Next or Step. A preference has been added to turn off JIT (Just In Time) compiling if this happens for other scripts.

- **Case 25365**

Character fields (e.g. the ALIAS/RE field on *MAT_RIGID) could not be set from JavaScript. Now fixed.

- **Case 25289**

Set.RemoveGeneralData() and Set.SetGeneralData() did not update the set cache for graphics properly so the wrong things could be drawn. Now fixed.

- **Case 25055**

Shell free edges were not available in JavaScript if PRIMER was run in batch mode. This is now fixed.

- **Case 24874**

The JavaScript function UnblankFlagged() did not work as expected for some entity types, such as Materials. This has been corrected.

- **Case 24684**

PRIMER could crash if the heading property for a joint was set without the id property being true. Now fixed.

- **Case 24659**

PRIMER had a global Error() method to write an error message to the dialogue box. Error is a reserved word in JavaScript and this is causing problems when used with some public JavaScript libraries. In version 12 the method has been renamed to ErrorMessage().

- **Case 24658**

The debugger could crash if a script contained large numbers of nested functions (functions inside functions). Now fixed.

- **Case 24639**

The JavaScript debugger could crash if a script had extremely long lines. Now fixed.

- **Case 24564**

When reading an LSTC dummy using the LSTC dummy conversion script, part of the dummy title could be lost. This has been corrected.

- **Case 24549**

Button clicks in JavaScript windows were not processed if modeless Edit(), Select() or Pick() was done from JavaScript. Now fixed.

- **Case 24528**

The exists property could not be set to true for *DATABASE ASCII cards. e.g. for model m, m.database.nodout.exists = true did not work. Now fixed.

- **Case 24519**

The select property did not get updated as a combobox widget selection changed. Now fixed.

- **Case 24483**

Toggling strict mode in the JavaScript debugger could cause PRIMER to crash when the script was run. This has been fixed.

- **Case 24029**

Deleting a model in JavaScript using Model.Delete() would make a modal JavaScript window become non-modal. Now fixed.

- **Case 23810**

If Widget.onTimer was used to hide a window as part of the callback then the script did not continue until the mouse was moved. Now fixed.

- **Case 23697**

It was not possible to obtain a label using the Set class function LastFreeLabel() in JavaScript if you wanted a label for a thick shell set, and there were no thick shell sets already in the model. This has been corrected.

- **Case 23678**

The JavaScript function FindSolidInBox() sometimes did not find all solids in the defined box. This has been corrected.

- **Case 23424**

If hover text was defined for a widget and '\n' was used to split the text into lines then it would cause problems if any of the lines were longer than 256 characters.

- **Case 23113**

Using try/catch did not work properly with the PRIMER JS API. It now does.

- **Case 21723**

A script could crash PRIMER if an object referred to an item in a model which had been deleted. Now the script will terminate with an error.

Selection

- **Case 25912**

Incorrect text could be displayed on mesh panel drop down buttons. This has been corrected.

- **Case 25081**

The following sequence of operations would cause (harmless) 'BadMatch' X11 errors on older Linux platforms, but could lock up PRIMER on more recent ones:

PART, Modify (or any other keyword)

Filter in object menu

Typing enough characters into the ascii text filter box to overflow it.

This was due to subtle errors in the GUI leading to a 'bad match' of windows on X11 windowing systems, ie Linux. It would not affect Windows platforms. Now fixed.

- **Case 24893**

When an object menu shows a single type of *ELEMENT, eg shells, for multiple models the an error in selection could occur in the following situation:

Menu shows results for this element type in more than one model.

One or more models contain > 10,000 instances of that element type.

One or more models contain < 10,000 instances of that element type.

In this situation all menu rows for all models will show a range of elements using [start .. end] syntax, and clicking on that row should select all elements in that start .. end range.

However if you clicked on the row of a smaller model with < 10,000 elements of that type you would only get the last element in that range (ie 'end') and not the full start .. end range.

The problem was that the syntax used for display was, to be consistent, always [start .. end] mode if any model contained > 10,000 items; but the parsing of a given row considered only that model. So if the model of that row had < 10,000 items the parser believed that it would be showing an explicit range of elements and picked only the last.

This has been corrected by making object menus 'remember' whether they are using explicit rows of elements of [start .. end] ranges, so that when a row is parsed it is treated correctly.

Sets

- **Case 25986**

PRIMER occasionally did not calculate the contents of _COLLECT sets correctly. Now fixed.

Shortcut

- **Case 24553**

Using the 'c' shortcut when the airbag reference coordinates and normal coordinates were swapped would swap the coordinates back again. Now fixed.

Sketch

- **Case 25038**

Sketching when moving the mouse over the clipboard row in an object menu could be very slow if the clipboard had a very large number of items in it. It is now significantly faster.

- **Case 22182**

If the sketch item was both blanked and off screen, the behaviour of single item (crosshair) sketch was not consistent with multi-item sketch in that a subsequent AC operation failed to draw the item into view space. Now fixed.

Text

- **Case 25500**

Using the 'text edit' function to edit Model comments or to text edit lines from the keyword editor could leave the temporary scratch file on the system.

This would only happen if the contents were unchanged, it was deleted correctly if the contents were modified. This was annoying, in that it could lead to a build-up of temporary scratch files, but did not stop PRIMER working correctly.

This is now fixed: these temporary files are now deleted regardless of whether they were modified or not.

- **Case 25297**

When using the Gvim editor on Linux as the editor for 'text edit' operations in PRIMER it is necessary to start it in 'no fork' mode so that it does not start a child editing process and then terminate the original launching process.

This is because PRIMER monitors the original process and takes its termination as meaning 'editing has finished' whereupon it inspects the file for any changes. Since the originating process terminates as soon as the separate editor is launched this means that text editing fails.

The fix is to launch Gvim using the '--nofork' argument, meaning that the PRIMER text editor preference should be set to something like:

primer*text_editor: /usr/bin/gvim --nofork

However this did not work because of errors in the way the text editor launched processes, it got confused by the extra argument.

This is now fixed: text editor strings can contain further arguments and these will not prevent their use so long as the original executable name, here /usr/bin/gvim, is a valid file on the system.

Timestep

- **Case 25430**

PRIMER did not calculate a modulus for MAT 57 correctly (for timestep calculation) if LCID was a table. Now fixed.

- **Case 24943**

Timestep for cohesive MAT184 elements is now calculated.

- **Case 24469**

Timestep calculation for MAT184 solids (cohesive elements) has been corrected.

- **Case 24336**

Timestep for MAT_169 when option extra = 3 has been corrected to include the effect of stiffness factors on the material card.

- **Case 23482**

Timestep of solids of formulation 16 was not calculated. Now fixed.

- **Case 23391**

Wrong time step for MAT_LOW_DENSITY_FOAM - Fixed now.

- **Case 19523**

Timestep for beams which use _OFFSET was not calculated. The option may effectively shorten the length of the beam and so decrease its timestep. Now fixed.

Undo

- **Case 24631**

When splitting shells PRIMER could crash if some of the shells were in sets and the sets had comments stored in them. Now fixed.

- **Case 21867**

'Undo' for node-based operations such as Drag and Replace, in conjuction with *AIRBAG_REFERENCE_GEOMETRY, was producing incorrect results. This has now been fixed.

Units

- **Case 25457**

Units conversion would fail for *BOUNDARY_RADIATION. This has been corrected.

- **Case 24632**

Units conversion of ln(strain_rate) was not being handled. Fixed this.

- **Case 24550**

Fixed: MAT_FABRIC TSRFAC (when -ve) x-axis units of curve should be Time (previously no units used).

- **Case 23871**

Unit conversion for LOAD_RIGID_BODY has been corrected to work with moments/rotation.

- **Case 23423**

The units for load curve LCID on *FRQUENCY_DOMAIN_RANDOM_VIBRATION optional card 6 were the wrong way round. Now fixed.

User interface

- **Case 25772**

A child Javascript window that has the 'on top' property set failed to show the popup menu associated with the top left [-] options button. It would flicker and disappear.

This was because that menu was not inheriting the 'on top' status of its parent window correctly, and thus getting demoted to lie behind it. Now fixed.

- **Case 25439**

In the *DEFINE_CURVE_SMOOTH editing panel it was possible to curtail the options shown in teh curve type radio button set by moving the bottom border of the window upwards to leave a very shallow region for the data entry buttons.

This was the that height, rather than the true available height in upper region where it was drawn, was being used to determine the space available for that radio button. Now fixed by using the correct value.

- **Case 25104**

PRIMER will now automatically start edit panels in modify mode if there is at least one item in the active model(s) [previously it was if there was more than 1 item in all models].

Additionally, if there is only one item in the active model(s) PRIMER will no longer automatically edit it.

- **Case 24501**

The interactive build panel has been resized and the layout changed to improve readability.

- **Case 23972**

When both HISTORY_NODE and HISTORY_NODE_LOCAL cards are present, the correct number of DATABASE_HISTORY_NODE cards is now reported in the menu.

- **Case 23960**

The *CONSTRAINED_JOINT editing panel has been redesigned to make it smaller and less confusing.

- **Case 23771**

All filename and pathname handling has been upgraded to handle strings of length 512 characters.

- **Case 23370**

In models with very many include files, the drop-down from model checking which enables filter by include file can run out of space. The function which assigns space to drop-downs has been improved to allocate space more cleverly.

Write

- **Case 25870**

PRIMER is capable of writing a 'd3plot' (or PTF) file of a model suitable for reading into post-processors. The size of this file was limited to 2GBytes, equivalent to roughly 50m elements.

This limit is now lifted and the file can be any size. In addition if the model to be written contains labels greater than the 32 bit integer limit the file will be written in 64 bit format.

- **Case 24686**

If there are two or more models in the database then Model, Write requires selection of the model to be output. This selection can be performed either by selection from a menu or by typing in a model number, and it works fine.

However if a selection is made and then the main PRIMER window is resized the currently selected model is lost. This is an inconvenience when running interactively, but can be a disaster when replaying a macro which assumes a current model has been selected.

The error was that a change in overall window size requires the Main (model) panels to be redrawn, and this redrawing process is also performed when the number of models in the

database changes. The two operations were considered to be identical, and assumed that the number of models had changed, thus cancelling any current selection.

The two reasons for redrawing have now been separated, and resizing the PRIMER window will no longer cancel an existing model number selection in the main Model panel.

2.1.2 Bugs Fixed in 12.1

Airbag

- **Case 27422**

PRIMER can now read/write & copy blank *AIRBAG_REFERENCE_GEOMETRY_RDT lines, so that the _RDT flag can be turned on for *AIRBAG_SHELL_REFERENCE_GEOMETRY

- **Case 26502**

PRIMER could crash if setting the NGAS field of an AIRBAG_HYBRID_CHEMKIN card to something greater than 10 during creation. This has been corrected.

- **Case 26209**

When changing an existing airbag definition from AIRBAG_HYBRID(_xx) to

AIRBAG_PARTICLE PRIMER attempts to map the old data intelligently onto the new. However, two omissions have appeared in this process:

Field HCONV on the HYBRID card was being interpreted as the wrong data type, meaning that its value was considered to be zero. This in turn meant that field NPDATA on the PARTICLE card would be zero when it might have been finite. This is fixed.

Field A23 on the HYBRID card was also misinterpreted, again resulting in a value of zero, meaning that the vent part id SID3 on the PARTICLE card would always be zero. This is fixed.

Attached

- **Case 28030**

PRIMER could crash if using 'find attached' with the tied contact one if the model contained a tied contact with a missing slave/master part ID. This has been corrected.

Belts

- **Case 27579**

When using Safety, Belts, New dummy to replace an old dummy with a new one the process could fail with an error if all the 'old' structure definition (ie the sets in the old dummy definition) contained elements that did not exist in the new dummy.

The reason is that 'replace dummy' has some functions that are common with the normal 'create seatbelt' operations, and when creating a belt it is illegal to try this if there is no structure to fit the belt to. This is checked and rejected, correctly, in the 'create seatbelt' case, but the same check was being applied to 'replace dummy' where having no structure is a reasonable condition at the start of the process.

This has been fixed by inhibiting this check in the 'replace dummy' case.

- **Case 27570**

If a deck containing *BELT cards after *END is hand-edited to remove both the segment set and the part set for the 'structure' to which the belt will be fitted then an auto-refit may fail with a label clash for *SET_PART.

This is because the part set is recreated, but the internally calculated 'next label to use' value for *SET_PART is not updated to consider the fact that this label is now in use.

Fixed by finding a new 'next label to use' in this situation.

- **Case 27443**

If:

- a seatbelt is fitted to a dummy that is in an include file

and

- this include file is referred to using *INCLUDE_TRANSFORM with label offsets

and

- the *BELT cards are themselves in that include file

Then the labels used for some of the sets that the belt fitter creates could lie outside the 'safe' range of labels for that include file, meaning that they become negative when the label offset is subtracted in order to write it out in its untransformed state.

This was due to a failure to consider label offsets when assigning the labels for new sets, and is now fixed.

(Note that -ve labels may still occur, for example if a node set on the belt includes a node label that is not in this include file. This will actually not invalidate the file, which will initialise correctly in LS-DYNA.)

- **Case 27397**

When fitting a seatbelt with 2d belt elements a stretch of straight belt between fixed ends (where 'fixed' means retractor, slipring or end point) should be treated as a simple straight line that is already in its final position. Instead, depending on the exact nature of the end points, the belt fitter could project these path segments outwards and pull them back to the straight line during fitting.

This was not a problem in itself, but because these were straight lines with no curvature the 'outwards' direction was indeterminate, meaning that such segments had a 50% chance of being projected the 'wrong' way.

That too did not matter, except that where a straight line between two sliprings was used, and the belt turned at an acute angle when passing through the slipring, the belt-to-belt contact could end up trying to depenetrate the belt in the wrong direction.

This has been fixed by making such segments 'fixed straight lines', with no need either to fit or to depenetrate. This presumes that the user has specified these lines correctly in their required final position.

- **Case 27048**

When an input deck containing *BELT cards had been hand-edited in a way that did not obey PRIMER's rules about [start/end] label pairs being contiguous, a crash could occur if the label range spanned was very large (10s of millions). This was because it was looping over the entire range making (literally) millions of latent definitions, and thus ran out of memory.

This has been fixed at two levels:

In V12 and V13 the 'loop over labels' logic has been replaced by 'loop over internal range', which avoids the creation of many unwanted definitions.

In V13 only an extra check has been added following keyword input that will detect this condition and warn the user, offering to condense the label range down if wanted.

This check will not only pre-empt the possible crash, but also warn the user that he may cause 'collateral damage' during belt fitting, since PRIMER assumes that the belt definition 'owns' all the items within a start/end label range, and will delete them. So if this range accidentally includes other bits of the model they too will be deleted, and this check should prevent that.

- **Case 26598**

Added a check to stop a crash in the belt fitter if a user tries to 'mesh' a belt that has no path points defined.

Case

- **Case 27411**

When there is a *CASE in a model PRIMER shows a [C] case management button at the top of editing panels to the right of the include file management button.

However the *CONTROL and *DATABASE editing panels are special cases in which this [include + case] button row applies to all cards, and not just a particular one, so showing the [C] case management button was misleading, and in fact clicking on it would either do nothing or give an error message - albeit a harmless one.

The [C] button no longer appears in these contexts, and some ergonomic problems which could cause the 'all sub-keywords' include file button to show the wrong status in some (rare) circumstances has also been fixed.

- **Case 27393**

When *CONTROL cards were placed inside *CASE statements PRIMER would read and process these correctly, but if the user decided to move these cards back into the main model, ie not in any case, this would not work.

This was due to internal errors when processing these cards, now fixed.

- **Case 27293**

When using *NODE_TRANSFORM or *PART_MOVE inside a *CASE definition the order in which the two types of transforms could go wrong, resulting in their failing to be applied or,

sometimes, being applied in the wrong directions. This was caused by internal confusion about the transformation status when building and changing *CASE status. It has been fixed.

- **Case 27241**

Errors were being thrown out when using control cards in conjunction with *CASE. These are now fixed.

- **Case 26378**

An error in the internal sorting of keywords inside *CASE meant that in rare circumstances data could get corrupted when moving back and forth between main model and backing store. The symptoms would be erratic, but would probably manifest themselves as spurious latent items. Now fixed.

Checking

- **Case 27337**

The check for lossy discretization of material curves could give false alarms when the curve passed through (but did not start at) y=0 as the tolerance of 2% of y value becomes meaningless. Now fixed.

- **Case 27251**

PRIMER could crash when checking *SET_SEGMENT_ADD and *SET_SEGMENT_INTERSECT cards. Now fixed.

- **Case 27249**

Model check applied to models which contained *LOAD_BODY and *CASE could result in a crash due to confusion over an internal type code. Now fixed.

- **Case 26964**

The calculation of timestep for resultant beams where section properties are defined by D1,D2,etc could be completely wrong due to a confusion between real and double precision variables in the calculation routine. Now fixed.

- **Case 26757**

The apply rules check function was found to SIGSEV. Now fixed.

- **Case 26325**

The test for table curves crossing was giving a spurious error for MAT83 case where TABID is < 0 and HU = 0. Now fixed.

- **Case 26223**

The check for slave nodes of a constrained contact which fail to tie because of a clash with another constrained contact was only working for cases where the master side was defined by shells. This has been corrected to work for any segment.

Connections

- **Case 27812**

When remaking subset of welds in a model with adhesive, PRIMER could give spurious error messages that spotwelder setup had failed and fail to update weld status correctly. Now fixed.

- **Case 27675**

When moving shells which have PRIMER connections attached between parts the option to 'fix replace' the layer definition could result in a connection of bad status for no apparent reason. Now fixed.

- **Case 27604**

The dropdowns on the connection table and create panel which list the xml filenames can display the absolute path. This may be excessively long and is now abbreviated when written to the button. The full name is given by hover text.

- **Case 27601**

Preference 'connection_xml_file_paths' which may be set to relative, absolute or dont_write (default) to control the filename that is written to any post-end connection which has been created by reading an xml file.

User who want xml filenames written must set this preference.

- **Case 27421**

On the connection table, the 'FE Info' column was not displaying the beam id for MIG beam weld connections. Now fixed.

- **Case 27409**

PRIMER now gives a warning message during creation of spotweld lines/adhesive if the layer information is inconsistent along the defined path.

- **Case 26917**

When redefining the path of adhesive lines or spotweld lines on the connections table, if you also had adhesive patches on the table, when remaking the connection the new path could be lost. This has been corrected.

- **Case 26810**

Master Connection Files containing adhesive line connections with only 2 points are now read correctly.

- **Case 26617**

When automatically creating PRIMER connection data from single solid spotwelds in a model, PRIMER would not take into account the solid local direction when forming the connection data. This meant that when the connection was remade, the local direction of the solids could be reversed. This has been improved so that the local direction is considered when forming the connection data. This only applies to single solid type spotwelds where there are 2 or more layers (i.e. 3T or more).

- **Case 26540**

PRIMER may refuse to automatically create a connection entity on a MAT100 solid/beam entity if any of the nodes on the solid/beam are referenced by non-structural entities like database history nodes. The restriction has been modified.

Constraints

- **Case 26881**

The constrained interpolation panel was found to SIGSEV if zero or an uninitialized parameter was entered into the INID field. Now fixed.

Contacts

- **Case 26496**

The units for the X axis for the UNLCID loadcurve field of *CONTACT_RIGID_TO_RIGID cards was velocity when it should have been length. This has been corrected.

Control

- **Case 26829**

*CONTROL_SUBCYCLE now opens the correct edit panel. Additionally, an issue which could cause the '+' button (used to add additional rows) to not work has been fixed.

- **Case 26538**

The include selection buttons for *CONTROL_ADAPTIVE did not work correctly. This has been corrected.

- **Case 26534**

*CONTROL_START was not being written out. Now fixed.

- **Case 26377**

Missing range autofixes for field FTALL on *CONTROL_CONTACT have been added

- **Case 26376**

The text box in CONTROL_CONTACT edit panel was not updated by drop-down for field FTALL. Now fixed.

- **Case 23719**

There were some control keywords that have filenames as fields but in PRIMER you could not input filenames containing spaces. This has been corrected.

Cut section

- **Case 27832**

The 'explain' help text referring to the capping of 2d shell elements in cut sections did not make it clear that selecting 'true thickness' would not, on its own, show the effect of any offsets of the shell's neutral axis as defined by NLOC on *SECTION_SHELL or *ELEMENT_SHELL_OFFSET cards.

Since normal PRIMER graphics shows shells as infinitely thin plates on the plane of their nodes, it looks confusing if cut sections show a plane that is offset from the uncut element.

PRIMER can visualise both true thickness and any offset, and this is done if the Display Options, Shell, True thickness option is set. If this is active then cut sections through shells also take into account any offsets, thus retaining consistency between the location of cut and uncut elements.

The help text has been updated to explain this more clearly, and the user manual has also had a short section added to provide more detail.

Database

- **Case 27159**

On edit, DATABASE_HISTORY entities were incorrectly moved from includes to the master model. Now fixed.

Deleting

- **Case 26715**

PRIMER could give an error message when performing a cleanup with a model containing assign mass definitions and an assign mass definition referring to an empty part set could be left in the model. This has been corrected.

- **Case 26565**

Cleanup unused did not work when selecting the master include file for cleanup. Now fixed.

Dummies

- **Case 27442**

Conventional dummy modelling in PRIMER assumes that a dummy will have a 'root' assembly, typically the pelvis, and that all remaining assemblies of the dummy will be children of this. This means that the computed angle of a joint between a parent and child, as defined by a constrained joint stiffness, will be based on the relative rotations of the coordinate systems on the parent and child sides, and - crucially - that those coordinate systems will be defined by nodes that are part of their respective assemblies.

However some dummy model builders have chosen instead to make the coordinate system on the parent side of the joint relative to a master system, meaning that reported angle of the child is 'with respect to master' and not 'with respect to parent'. This tends to be done for the torso and

head assemblies, making it easier to correlate their angles with some master system that is effectively part of the seat.

PRIMER will handle this, but with two undesirable side effects:

(1) During input, and subsequently in a model check, the fact that the coordinate system on the parent side is not attached to the parent is reported as an error.

Strictly speaking, given the rules stipulated for dummy modelling in PRIMER, this is true. However those rules never considered the 'relative to master' way of working, and to report this as an error is over-conservative since the rules are too restrictive.

Therefore in release 12.1 these errors have been demoted to warnings. In future release they will be removed altogether once the necessary extra checks that the master system is genuinely attached to the dummy have been written.

(2) When positioning dummy assemblies in 'rotate angles' mode it is possible to type in a new angle about some degree of freedom, and for the dummy to move accordingly.

This works correctly for the 'relative to master' logic described above, but because PRIMER assumes that all angles within a dummy are 'relative to parent' is does not recalculate the reported angles of any children of the assembly which has been moved, and therefore the reported angle in the table does not change. It is however updated correctly by any action which results in all angles in the dummy being recomputed.

This has now been fixed: changing the angle of any assembly now forces a recalculation of the angles of all assemblies in the dummy, which will pick up any consequential changes of angle at joints which are 'relative to master'.

- **Case 26618**

For the Dyna Dummy Positioning feature, when selecting 3 nodes per assembly, there is an option to 'Automatically determine nodes'. This would choose the wrong nodes if the user had pre-selected to rigidify some assemblies. This has been corrected.

Edit

- **Case 27830**

The Text Edit button on the editing panel for *DEFINE_TARNSFORMATION was greyed out (not available) unless some data fields had been edited on the card or the card was fully populated with data and thus valid for create/update. The text Edit button is now unconditionally available.

- **Case 27829**

PRIMER permits creation of items, e.g. elements, to be 'daisy-chained' via popups. For example when editing an element a right-click on its PID field, 'create', will map a panel to create the part in question. Similarly a right-click, 'create' on an item field in the keyword editor will create that item.

This all works OK, except that some editing panels (nodes, elements, database cross-sections and some commonly used constrained items) have all been modified to default to work in 'auto-create' mode, where each time something is created the panel is reset and automatically moves on to create the next such item.

This causes problems in the daisy-chained context since there is only a single 'slot' into which the new item can fit, and attempting to create more than one such item is both pointless and, in the case of the keyword editor, may generate internal errors.

This behaviour has been fixed: daisy-chained creation will now ignore any auto-create settings, and the panel will self-dismiss after the first such item has been created.

- **Case 26921**

When editing a scalar *CONSTRAINED_LAGRANGE_IN_SOLID card in which field NVENT is > 0 there are an open-ended number of extra rows of vent coefficients that can be edited, and these are handled by PRIMER's standard scrolling list. One of these fields, CEOFFLC, is a loadcurve, and this was not being processed correctly meaning that if the right-click popup SELECT option was used to select an existing loadcurve it would crash. Now fixed.

- **Case 26560**

The joint panel was redesigned for version 12. An error in the redesign meant that node numbers were not shown in the panel when nodes were picked. Now fixed.

Element quality

- **Case 27616**

Element warpage was not calculated correctly for shells that had internal angles greater than 180 degrees. This has been fixed. Additionally, a variable representing scale factor in the quality optimisation algorithm was not being reset after multi-node optimisation operations. This could have led to corner nodes not being optimised in rare cases and has now also been fixed.

Elements

- **Case 26841**

The *ELEMENT_SHELL_NURBS_PATCH editing panel did not work correctly. Now fixed.

Error trapping

- **Case 27740**

If a deck contained an *ALE_AMBIENT_HYDROSTATIC card with an invalid stype field, the error message given was incorrect. Now fixed.

- **Case 26625**

PRIMER could give an error message when reading a *TERMINATION_BODY card. Now fixed.

- **Case 26624**

If a deck containing a *CONTROL_IMPLICIT_BUCKLE card with a BCKMTH field, was written in a version less than R7.1, there was an error in the logic that culled the field that could cause a crash. Now fixed.

FMH

- **Case 26551**

PRIMER could refuse to read post *END headform data in a *POSITION card if the field was blank. This has been corrected.

General

- **Case 26568**

A crash occurred on Linux during screen-picking, and from the trace-back it seemed likely that it was due to threaded screen-picking.

The cause is not known precisely, and we have been unable to reproduce it. Based on the evidence seen so far a tentative fix has been made to prevent what is - at least theoretically - the problem.

- **Case 26309**

In rare circumstances the listing of checkpoint files (following a previous abnormal termination) could show the same file multiple times. This doesn't matter, and there is no effect on the overall functionality of the code, but it might look a bit confusing, and if all files are chosen for deletion a warning message will be printed when an attempt is made to delete the file for the 2nd time. Now fixed.

Graphics

- **Case 28007**

Attached node labels were left behind by animation of BOUNDARY_PRESCRIBED_MOTION. Now fixed.

- **Case 27789**

When a model containing (IGES or other) 'surfaces' is drawn in shaded mode these use 'smooth' shading to give the appearance of continuous curvature across the surface. This all works well, but a problem can arise if the same model also contains ordinary (element) parts that have been made transparent, since these parts will suddenly swap from 'transparent' to 'opaque with a strange colour' if surfaces are also drawn in the same plot. This will only happen if hardware shaders are being used, which is the default on modern high-performance graphics cards.

The problem is due to an obscure error in the graphics driver when it swaps from 'smooth' shading for the surfaces, back to 'flat' shading for the parts, and will only appear if these parts are

also transparent. This only showed up when the ability to fit a surface to a mesh was added during V13 development, and it is highly unlikely that it will be encountered in earlier versions, nevertheless it has been fixed in V12.1.

- **Case 27775**

The graphics function Node Plot(SPC) was not correctly plotting the case of a rigid part with local constraint (CMO = -1) defined on the MAT20 card. Additionally, global constraint could be incorrectly plotted when CMO = 0. Now fixed.

- **Case 27761**

When Beam or Shell 'true section' drawing is in use then these element types not only show their true thickness but also any offset defined by a range of means on the element cards themselves, or on the section cards.

Updating *ELEMENT cards caused the elements to be redrawn using the new attributes, but updating the *SECTION card to change attributes such as thickness or NLOC was not forcing a redraw of these elements in their new size or location. Now editing a *SECTION card will force a graphics refresh of SHELLs or BEAMs if either is being shown in 'true thickness' mode.

- **Case 27388**

Using the 'k' shortcut reinstates the visibility switches of all items in the model, turning them all on. In some cases the sequence 'k' followed by 'a' to autoscale the image might not calculate the correct scale, resulting in only a subset of the model being autoscaled correctly.

This is because the 'k' shortcut was not resetting the internal requirement for the bounding box around the model to be recalculated. Now fixed.

- **Case 27200**

*BOUNDARY_PRESCRIBED_MOTION_RIGID_LOCAL was being drawn in the global axis system (rather than the local system) when the rigid part was a Nodal Rigid Body. Now fixed.

- **Case 26912**

PRIMER did not take into account multiple cut-sections when using AC of doing a visible contact penetration check. This has been corrected.

- **Case 26889**

If 1d seatbelt elements form a straight line, the algorithm that calculates the normals to enable them to be plotted as 2d elements could become ill conditioned, giving an untidy plot. Now fixed.

- **Case 26696**

When a dynamic viewing operation changed the centre of rotation, or the image scale, then while the screen (x,y) coordinate about which subsequent rotations occurred (typically the screen centre) was known, the screen z coordinate (depth) of this point was not, and needed to be recalculated.

In addition in order to speed up screen picking PRIMER caches the 2d screen space coordinates of nodes, and a dynamic viewing operation which changes these invalidates this cache, requiring it to be recalculated.

Both these operations were carried out correctly and successfully, but in some circumstances they took place when a new dynamic viewing operation was initiated, and when a model is large (eg >5m elements) this could cause a significant delay between the initial movement of the mouse and a subsequent change of view.

The sequence of operations has been changed so that these expensive calculations now take place immediately after a viewing operation that requires them has finished, thus using the 'dead' time between the user finishing one mouse operation and starting another. In addition they have been rewritten with parallel processing to make them faster.

Some other viewing bottlenecks have also been fixed:

- Using the Control key means that only the wireframe elements of the image are shown, but releasing this key was slow because the full image was redrawn fully. This was not necessary, and it is now redrawn from the graphics cache giving a near instant response when the control key is released.

- Using control alone to give wireframe, or shift+control to give free edges, also presented problems when 'Predictive picking' (which shows what is under the cursor) was active. Any movement of the mouse in these modes that caused a change in the item under the cursor triggered a redraw in the original plotting mode, therefore the wireframe or free edge image would revert to shaded as soon as the mouse was moved. This is now fixed and the item under the cursor will be shown in the current drawing mode.

- **Case 26646**

In both PRIMER and D3PLOT you can use the control key (ctrl) to make the image revert to 'wireframe overlay only', and shift+ctrl together to draw temporarily in free edge wireframe mode.

This worked, but the shift+ctrl case could be slow to start dynamic viewing on Windows platforms because it redrew the image up to 7 times before moving the image.

This was due to the command handler on Windows generating spurious extra 'shift' and 'control' key events, making the application think that these keys had been pressed several times. This was a (wrong) fix for the problem that pressing shift or ctrl while the mouse was in a different window might otherwise not be 'seen' by this application. It has been fixed by coding a more rigorous test.

- **Case 26549**

PRIMER could very occasionally get stuck in screen area mode if you dragged a small distance on the screen. Now fixed.

Help

- **Case 26410**

Help button on main panel now works for SENSOR and RAIL cards.

Include

- **Case 27839**

When using multiple *INCLUDE_PATH definitions, if one of the include paths was a substring of another include path, PRIMER could write out relative include paths incorrectly in master files. This has been fixed.

- **Case 27470**

Performing an 'Include Add' or 'Include Replace' from the include tree on an include file that contained *NODE_TRANSFORM cards could cause the transforms to be applied twice, resulting in the affected nodes moving to the wrong location. Caused by internal confusion about when to apply transforms, now fixed. (Ordinary 'read' of a deck including such an include file worked OK.)

- **Case 27036**

When writing the filename after *INCLUDE, if the total length exceeds 80 characters, PRIMER will attempt to write the name on a separate line below the directory. If the directory length (with name removed) was exactly 79 or 80 characters, the continuation character ' +' got written just beyond the 80 character limit which is illegal in LS-Dyna. Now fixed.

Include transform

- **Case 27571**

PRIMER sometimes did not handle compound *INCLUDE_TRANSFORM files with transforms correctly when the *DEFINE_TRANSFORM card was in a different *INCLUDE file. Now fixed.

- **Case 27465**

Two problems have occurred with 'Replace include' if the include file being replaced is inside an *INCLUDE_TRANSFORM.

(1) PRIMER could crash. This was due to the old file not being completely purged from the database, instead leaving behind some references to a file that not longer exists.

(2) The name of the new file was not being updated correctly, leaving the 'old' name in the include tree and - worse - also in the keyword output file. Due to a failure to update the new name in the right places.

- **Case 26877**

PRIMER would not read an include transform definition if it contained multiple *CONSTRAINED_NODAL_RIGID_BODY cards with the same PID. Now fixed.

Javascript api

- **Case 28022**

The javascript part function which gives Dyna_Part_Mass had an uninitialized variable leading to junk values for slaved rigid parts. Now fixed.

- **Case 27781**

The local property was not available in the Beam class. It has been added.

- **Case 27780**

Loadcurve IDs higher than 999999 could not be set for loadcurve properties (e.g. lcid on *ELEMENT_DISCRETE_LCO). Now fixed.

- **Case 27756**

If there was no composite data in a shell then calling Shell.SetCompositeData() would cause a crash. Now fixed.

- **Case 27524**

When setting a function for *DEFINE_FUNCTION in JavaScript '\n' characters can now be used in the string to say where the string should be broken into lines.

- **Case 27424**

Blanking and unblanking of sets in JavaScript did not work the same way as doing it interactively. It now does.

- **Case 27238**

If a default value was given for Window.GetInteger or Window.GetNumber and the user just pressed OK rather than giving a new value the value was not returned. Now fixed.

- **Case 27225**

The RigidBodies class did not have a label property. This has been added as even though there is no LS-DYNA label for *CONSTRAINED_RIGID_BODIES cards PRIMER still has an internal label that can be used by scripts.

- **Case 27218**

Custom JavaScript checks did not work for the RigidBodies and ExtraNode classes. Now fixed.

- **Case 27205**

When using the JavaScript Model class function Attached() PRIMER should flag everything found that is attached to items originally flagged. This did not work correctly when the recursive flag was on. This has been fixed.

- **Case 26641**

Blank lines in the message argument for Window.Information() were incorrectly removed. They are now retained.

Keyword

- **Case 28000**

For *AIRBAG_SHELL_REFERENCE_GEOMETRY you had to specify a PID on the input line to be able to click on the '+' to add it into the main list. This is an optional field, so you should not have to specify it. Now corrected.

- **Case 27835**

*SENSOR_DEFINE_NODE(_SET) can now defined without VID which is optional.

- **Case 27782**

PART_MODES can now be created without ANSID which is optional.

- **Case 27567**

NDSET in CONTROL_FORMING_SCRAP_FALL was read as a part label when it should be node set. Now fixed.

- **Case 27490**

PRIMER would not read *INITIAL_STRESS_TSHELL when LARGE=1. Now fixed.

- **Case 27432**

*BOUNDARY_ELEMENT_METHOD_NEIGHBOR cards are now read in and written out correctly.

- **Case 27255**

PRIMER now supports the environment variable KEYIN_NO_SMALLLARGE_INHERIT which, when set to 'true', will permit keyin of mixed format decks that have a long-format master and short-format includes. Note that mixed-format decks are not supported in LS-DYNA as of 971R7.1.2.

Additionally, 'long=s' tags will be written to *KEYWORD for short format includes as long as the master is in long format.

- **Case 27252**

PRIMER was not reading *CONSTRAINED_COORDINATE. This has been fixed.

- **Case 27250**

LOAD_SURFACE_STRESS cards now write out correctly when using large labels.

- **Case 27180**

SECTION_SPH suffices could be written incorrectly on keyout. Now fixed.

- **Case 27072**

SENSOR_DEFINE cards can now be edited from drop-downs model modified panel as well as part tree menu.

- **Case 26782**

The category DEFINE_VECTOR was lost on the entities panel. Now fixed.

- **Case 26692**

The rigidwall keyword editor did not work correctly for geometric rigidwalls. Now fixed.

- **Case 26604**

*INITIAL_STRESS_SPH fields are now written out in the correct order.

- **Case 26566**

Models containing ELEMENT_SOLID_DOF cards were previously not being read by PRIMER. This has now been fixed.

- **Case 26559**

On Create/Modify panel for ELEMENT_SEATBELT_RETRACTOR, SID3/SID4 input has been corrected.

- **Case 26558**

Keyword editor was losing fields for IXX -- IZZ on *DEFORMABLE_TO_RIGID_INERTIA. Now fixed.

- **Case 26522**

PRIMER could crash if a *CESE_PART is selected for the PID field when creating a *BOUNDARY_PORE_FLUID card. Now fixed.

- **Case 26382**

PRIMER would not read the header options for *CONTROL_FORMING_ONESTEP and

*CONTROL_FORMING_OUTPUT. This has been corrected.

- **Case 26320**

There was an error in the keyword reader that would result in the card

*LOAD_VOLUME_LOSS not being read. Now fixed.

- **Case 26217**

Incorrect invalid context warnings are no longer displayed if functions are used to define load-curves for various *LOAD_SEGMENT cards.

Keyword editor

- **Case 27899**

PRIMER could crash if changing a thermal material type or equation of state type of the keyword editor of those cards. This has been corrected.

- **Case 26836**

When recording a macro in a keyword editor button presses recording for a suffix may not have been recorded properly if there were multiple suffices for the keyword (the wrong suffix may have been recorded). Now fixed.

- **Case 26834**

Errors in HMIN, HMAX and SPHINI fields in the keyword editor for SECTION_SPH have been corrected.

- **Case 26527**

Fields VT/VH on keyword editor for *BOUNDARY_SYMMETRY_FAILURE corrected.

- **Case 26212**

When using the keyword editor to process items with many rows of data, typically *INITIAL cards, the update of the editor become quite slow when rows were scrolled or any global edits were made.

This was because an expensive update was taking place unnecessarily every time a data field on a row was drawn, rather than once per row. Now fixed.

Labels

- **Case 26818**

The final block of labels for an entity type in the visualise label panel would not be shown if the last entity in the label list was latent. This has been corrected.

- **Case 26622**

When ELEMENT_BEAM_THICKNESS which has discrete section was read before its SECTION card, the label of the DEFINE_COORDINATE_SYSTEM risked being lost if it was a long label (greater than 6 digits) as it was stored in a float. This has been fixed by allocating a 64 bit integer slot.

- **Case 26399**

When entering a floating point value into Material editor fields that could be either floats or labels (typically -ve = item label, +ve = fixed floating point value) a spurious warning message about an invalid integer value could be written when a float is input.

This doesn't actually matter since the data field is read correctly. It will typically only occur if the float is given without a leading digit, ie '0.4' will parse silently, but '.4' will produce the error message. Now fixed.

Macro

- **Case 27617**

Playing an old macro in v12.0 relating to the belt fitter could fail. This has been corrected. Also, running two macros (carrying out belt fitting) after each other, and deleting a model in-between could cause a crash. This has also been corrected.

- **Case 27292**

The 'Offset by part thickness' option on the 'project to mesh' panel under 'orient' could not be reliably turned on/off via a macro. This has been corrected.

- **Case 27186**

When recording a macro if button presses in menus were recorded when the menu had automatically expanded the macro may not have played back properly if there was more than one menu open in PRIMER with the same name (e.g. 'FILTER menu'). This has been fixed.

- **Case 26645**

Macros could play back button press commands in windows in which input was banned. Now fixed.

Manual

- **Case 27227**

Keyword 'K' shortcut button now opens manual at either *ELEMENT_SHELL or

*ELEMENT_SEATBELT depending on whether SEATBELT option is invoked in Shell Create/Modify panel.

Mass

- **Case 27591**

When using Remove->Cleanup on a model containing assign mass definitions that referred to an empty part set, PRIMER could give error messages and not delete the empty set and assign mass definition as you might expect. This has been corrected.

- **Case 26133**

Mass of a 2d seatbelt element part has been adjusted to be 1/10th of the sum of element masses. This will give agreement with the values reported in d3hsp file.

- **Case 19471**

Calculation of added mass for MAT100 solid spotwelds which use non-zero DT has been modified to agree with results reported in LS-Dyna d3hsp file i.e. to adjust the density of the solid spotwelds and increase their structural part mass.

Materials

- **Case 27610**

MACF on orthotropic materials can now be 4 where previously only values or 0 through 3 were permitted.

- **Case 27040**

PRIMER could crash when opening the edit panel for MAT_59. This has been corrected.

- **Case 27024**

The field TSRFAC on *MAT_FABRIC is defined as follows:

< 0.0 is the label of a loadcurve of one sort

0.0 - 1.0 is a floating point value

> 1.0 is label of a different sort of loadcurve.

Values in the range 0.0 to 1.0, which should have been treated as scalar

floating point values, were not being processed correctly and would just revert to 0.0. The +/- loadcurve options still worked corrected.

This worked correctly in PRIMER 11, but the problem appeared in 12 as result of changes to process 'large' labels. Now fixed.

- **Case 26745**

*MAT_ADD_EROSION does not accept LCSRS as a negative ID. Now Fixed.

- **Case 26391**

RBCFAC in MAT187 is now a float rather than a loadcurve.

- **Case 26237**

MAT_233 E can now be -ve loadcurve.

- **Case 26233**

MAT179 LCID1,LCID2 can now be tables

Measure

- **Case 27578**

Changes have been made to speed up part to part measure.

Memory

- **Case 26629**

Model with LOAD_SEISMIC_SSI may crash as memory allocation for the keyword was incorrectly programmed. Now fixed.

Menus

- **Case 27208**

Wildcard format processing has been fixed for Key In text box in object menus.

- **Case 26886**

PRIMER could crash in the CONTROL_MPP_DECOM_ARRANGE_PARTS panel if changing one of the rows of data from one type to another (part to part set), then trying to edit the item. This has been corrected.

- **Case 26739**

If PRIMER (or any other application) was activated by clicking on the window preview icon in the Windows task bar, the main application window was raised but any other windows that had been dragged onto the desktop were left hidden.

- **Case 26392**

When the Display Options 'True sections' button was ticked on the beam display popup a message about 'Error in sm_modify_button' would appear in the terminal window. This was harmless, and was the result of trying to update the status of a non-existent button, preserved for backwards compatibility with macro files written in PRIMER V11. The Warning message has been removed.

- **Case 23074**

Rarely, and it depended on the screen resolution and the setting of the 'Display factor' under Menu attributes, the right hand side (RHS) Keywords and Tools area could swap itself from the default of four columns of buttons to three. Now fixed.

Merge

- **Case 27540**

It was sometimes difficult to see the low:high numbers on the model merge model data panel. This has been improved.

- **Case 26478**

Errors in model copy for *DATABASE_ENVELOPE, *DATABASE_MASSOUT, *DATABASE_PAP_OUTPUT cards have been fixed.

Mesh

- **Case 26705**

Combining shells did not work correctly if the shells were in a set. Some of the shells were not deleted. Now fixed.

- **Case 26676**

Splitting shells could create latent shell definitions. Additionally, if a latent shell was split PRIMER could crash. Both these problems have been fixed.

- **Case 26649**

Models containing large part labels for various volume III mesh cards can now be copied correctly.

Model build

- **Case 27754**

Upper leg type (PEDLEG_UPPER) model build operations could provoke a crash in PRIMER if a non-existent CONTACT were specified in the build panel. This has been fixed.

- **Case 27627**

In Model Build the 'Build from CSV file' option allows you to assemble models from component include files by using commands and coordinates in a CSV file. For some cases you can also defined a *PARAMETER name and value, allowing predefined references to this parameter to have their values updated on a per-loadcase basis during the build.

There were some errors in the internal treatment of this parameter which could lead to it not being applied correctly in all cases, and in addition any other (unrelated) parameters in the model could get deleted in some cases. The exact behaviour would depending on the options chosen for the build. The parameter is now applied correctly.

Model modified

- **Case 27654**

The default on the 'what's modified?' tree viewer has been switched to show items moved between includes as a model difference. This has been done so that the info on the tree view will accord with the red lighting of includes when modified is pressed on the include tree.

- **Case 27574**

PRIMER would sometimes print property, geometry and keyword difference information for parts in the model modified tree in an inconsistent way. This has been corrected.

- **Case 27297**

Copying and deleting *SET_COLLECT definitions could in certain circumstances lead to corrupted SET definitions. This has been corrected.

- **Case 27290**

PRIMER could fail to compare models containing *ICFD_MAT during a model modified operation. This has been corrected.

Nastran

- **Case 27655**

Wrong number of CBUSH elements were being written out when Nastran output option 'Convert MIG to RBE3 with CBUSH option' enabled. Now fixed.

- **Case 26652**

PRIMER tries to read PLOAD4 cards as PLOAD cards when reading NASTRAN. Now fixed.

Operations

- **Case 26371**

The internal 'flagging' routine for card type *NODE_TRANSFORM was left out of an internal list of functions.

The effects of this are hard to quantify since flagging is used for so many internal operations in PRIMER, and they may be far-reaching.

This is a serious error and users should either try to avoid using this card, or at the very least check carefully that it has been processed correctly following a sequence of operations.

The issue has been fixed in 12.1

Orient

- **Case 27403**

PRIMER did not orient *AIRBAG_REFERENCE_GEOMETRY and this could cause problems. There was special logic to ensure that *MAT_FABRIC axes were kept consistent with airbag geometry. PRIMER now does orient reference geometry and *MAT_FABRIC axes. This behaviour can be changed with 2 new options for orient.

- **Case 27391**

Copy orient was incorrectly copying the ELEMENT_MASS_PART(_PART) card when in 'use same part' mode. Now fixed.

- **Case 27050**

Orient-Project to mesh operations could fail under rare circumstances as a result of the location of shells at the edge of the target mesh. This has now been fixed.

- **Case 26746**

Orient->Project menu buttons were not being remapped after the Options sub-menu were opened and dismissed. This has been fixed.

Output

- **Case 27371**

When writing *SET_SEGMENT_ADD and *SET_SEGMENT_INTERSECT SOLVER was incorrectly written in the 6th field instead of the 2nd. Now fixed.

- **Case 27052**

Two new keyout options and corresponding preferences have been added to support long/mixed format decks:

- 'Keep existing' will attempt to preserve the current format for each include and the master file

- 'Automatic' will write includes (and master) containing large labels in long format (20 char wide fields). All other includes will be written in the normal short format. Previous formats will be ignored.

A now-defunct format (L_80_C) that'd write long format cards over multiple rows has now been deprecated but can still be accessed using an environment variable.

Pdf

- **Case 26351**

Editing windows for control cards accessed via Edit buttons on the main control panel now have a keyword manual shortcut link.

Parameter

- **Case 27433**

In extremely rare circumstances (this has only ever been seen once in an internal test model) parameter definitions could get wrongly assigned at random to items.

This was due to internal read and/or copy operations leaving an internal list of parameters used during copy operations still 'populated', meaning that a further create or copy operation might wrongly pick this list up and assign parameters to fields on the items being created or copied. This would only happen in extremely unusual circumstances and, so far as we know, has never happened outside our normal testing process. The internal parameter copy buffer is now cleared more thoroughly.

- **Case 27401**

If a parameter was used in multiple transformations and it was changed the model may not have updated correctly. Now fixed.

- **Case 26263**

There was a bug in the editing panel for *AIRBAG_REFERENCE_GEOMETRY_BIRTH which meant that if the birth field was defined using a parameter this would get 'lost' when the card was edited.

A further bug, affecting the editing of both Airbag REFERENCE_GEOMETRY and SHELL_REFERENCE_GEOMETRY was also fixed. If there were 2 or more models in the database the editor could sometimes get confused about which model it was dealing with.

The above 2 fixes have gone into V12.1.

In addition to the above the following have been added to V13:

- A Check function for both Ref Geometry and Shell Ref Geometry

- A further check for the (now) known bug that LS-DYNA will assign the same 'birth' time to all reference geometry definitions even if they differ or have none. There is a fix for this from LS-DYNA 971R1 on the *MAT_FABRIC card.

- **Case 26252**

A problem can arise if a deck contains:

include file using parameters

*DEFINE_TRANSFORMATION cards referencing these parameters

*INCLUDE_TRANSFORM referencing the above

This works when cards are defined in the order above because the include file containing the parameters precedes the cards that use them. However after reading this model into PRIMER and writing it out again it fails to initialise in LS-DYNA because PRIMER has re-ordered the keywords to the following order:

- *DEFINE_TRANSFORMATION cards referencing these parameters

- *INCLUDE_TRANSFORM referencing the above

- An include file using parameters

Meaning that the parameters are 'referenced before they are defined', and

LS-DYNA will reject this (although PRIMER will reread it correctly).

This has been fixed by adding an extra pass in models containing parameters that does the following:

Looks for *DEFINE_TRANSFORMATION and/or *INCLUDE_TRANSFORM cards that reference parameters.

Looks to see if these parameters are defined in an include file, rather than in the master file.

If this is the case it writes out the include file(s) containing the parameters before the two cards which reference them.

Part tree

- **Case 27405**

With the 'part tree blanking status' turned on, it was difficult to see when only some parts in includes where blanked/unblanked. The logic has been changed so that red/green bars used to display this feature.

Pedestrian

- **Case 28008**

The pedestrian markup script was labelling upper and lower leg points incorrectly. It was labelling the left hand side with positive numbers and the right hand side with negative numbers. It should have been the other way around. Now fixed.

- **Case 26401**

The angles for marking up the bumper lines could not be changed from the defaults. Now fixed.

Preferences

- **Case 26276**

The default pdf and html applications used by PRIMER can now be changed on windows with a preference.

Quick pick

- **Case 27404**

When colour, transparency, or drawing mode is set for an include file by quick pick, the part properties are backed up so that subsequent undo operations will restore the colour, etc on a per part basis.

- **Case 26698**

The quick pick function Keyword for material does not bring up the keyword editor multiple times anymore.

Read

- **Case 27928**

When switching to IGES mode the extension assigned to the filename was written as .ig*s when it should be .igs. Now fixed.

- **Case 27824**

*PERTURBATION_NODE with TYPE=4 and CSTYPE=5 or 6 would not read into PRIMER correctly. This has been corrected.

- **Case 27450**

If *INCLUDE was followed by a directory path but no filename, PRIMER would SIGSEV as it expected a name for the include. This will now return an error code which terminates the read.

- **Case 27153**

PRIMER would not read the PID line correctly for *ELEMENT_BEAM_PID if OFFSET or ORIENTATION option was set and the PID's referenced were over 6 digits long. This has been corrected.

- **Case 26685**

The file drop method for reading a keyword file on windows (a) failed to update the error and warning count on the read panel (b) failed to build the data needed for the part tree display to work. Both these are fixed.

- **Case 26562**

If data on a *DEFINE_TRANSFORMATION card was misaligned so a zero 'spilled over' into the next field PRIMER did not read the data correctly. The error trapping is now better.

- **Case 26379**

An error in the keyword editor for *SENSOR_DEFINE_NODE has been fixed.

The field NODE1 would be shown wrongly in the editor (despite being read correctly from file) and attempts to modify this field would also fail. Caused by internal confusion trying to distinguish between the NODE and _SET variants, and ending up rejecting both!

- **Case 26374**

PRIMER would give an error if N2=N3 for a beam but LS-DYNA would initialise the model without any errors. The error check has been removed from PRIMER.

- **Case 26335**

Keyword *USER_LOADING (no suffix) would not be read due to an error in the keyword reader. Now fixed.

- **Case 26334**

The _SET suffix on the *SENSOR_DEFINE_NODE card would be ignored on keyword read. Now fixed.

- **Case 26324**

PRIMER would refuse to read *INITIAL_STRESS_BEAM/SOLID cards which has NHISV set but LARGE was not set to 1. This is not a valid input, but it has been created in some DYNAIN files. PRIMER will not set NHISV to zero when reading if LARGE is zero.

Renumber

- **Case 26933**

In the renumber selection panel it could be possible to renumber entities and then have items with duplicate labels. This has been corrected.

- **Case 26896**

Renumber includes > CLASH > Information now also outputs clash information if there are only General clashes in the model.

Rigidify

- **Case 27173**

The dropdown for selecting or creating a rigid material on the rigidify panel was causing a crash. Now fixed.

- **Case 26856**

PRIMER would not process a label correctly when using the RIGID_MATERIAL command for command line rigidify. PRIMER would also not process long labels in dialogue box commands properly. These have been corrected.

Sections

- **Case 27197**

When true beam sections are displayed for beams which have their section properties A, Ixx and Iyy explicitly defined, PRIMER tries to draw them as an equivalent I-section. If a model was defined using metres as the length unit sometimes it wouldn't be drawn correctly. Now fixed.

Selection

- **Case 27166**

If an out of range model number was specified in Key_In for object menus a crash could result. Now fixed.

- **Case 27158**

When performing mouse picks of multiple items, eg nodes, in a large model and combining this with rapid dynamic viewing and other mouse movement, the picking could get confused and stuck in a spurious 'pick by dragging out an area' mode, with the start point of the area in a wrong position. Actually finishing the operation by dragging out the area would cure the problem, but it was annoying because it would not have been what you intended.

This was caused by the rapid series of mouse events, combined with a busy cpu processing the results of these, leading to things occasionally being handled in the wrong order. It has been fixed by a more robust way of handling these rapid events.

- **Case 26480**

PRIMER could crash if filtering *DEFINE_CURVE by 'not referenced' in object menus. Now fixed.

- **Case 26170**

Selection menus ('object menus') in PRIMER will, by default, expand automatically if they are not large enough to show the relevant contents, and then they will shrink back to their original size when the mouse is moved out of them. This works correctly and has not been changed.

They will also shrink themselves if any of the 'Key in', 'Filter' or 'Vis' options are used, since these are mapped alongside the menu and would be obscured by it in its enlarged state. This too works correctly, and has not been changed.

However the user can also drag a selection menu out of its original parent window to be 'floating' within the main PRIMER window, or he can also drag it completely out of the PRIMER window onto the desktop. In these situations the original behaviour would still have been to shrink the

menu back down either when the mouse moved out of it, or when any of the 'Key in', 'Filter' or 'Vis' buttons were pressed, and this was perverse because there would be no clash.

Therefore if a selection menu has been dragged clear of its parent, or onto the desktop, it will no longer be 'shrunk back' in these situations; this behaviour will only recur if the user drags it back to its original position within its original menu panel.

- **Case 26164**

When toggling model visibility using up/down arrows, or setting model visibility explicitly in the Model, List menu the 'tabs' on top of open object menus did not also change.

Model 'tabs' on top of open object menus will now update to the currently visible model(s) as model visibility is changed. This also means when model visibility is toggled from model A to B it will be possible to screen-pick newly visible items from model B.

Sets

- **Case 27476**

PRIMER could give error messages when doing some operations if the model contained *SET_MODE_LIST_GENERATE cards. Now fixed.

- **Case 27399**

PRIMER would crash when editing/adding information to a *SET_SEGMENT card with the _ADD or _INTERSECT flag on. This has been corrected.

- **Case 27242**

Incorrect warning about use of _GENERATE in SET_DISCRETE and SET_TSHELL has been removed.

- **Case 27230**

Wrong error messages for combinations of _COLLECT with other suffices for sets have been removed.

- **Case 27161**

*SET_COLLECT sets with the same label were always treated as clashes in model merge. A new option now allows them to be merged if required.

- **Case 27160**

When carrying out a 'replace include' operation, if the include being replaced contained *SET_COLLECT definitions, the definitions could be corrupted. This has been corrected.

- **Case 26732**

PRIMER could crash when deleting a *SET_MODE card. Now fixed.

Sketch

- **Case 26541**

The sketch button for CONTACT_COUPLING was causing crash. Now fixed.

Standard templates

- **Case 27801**

The standard templates for pedestrian head and leg impacts would fail if it was run on Windows and the .LST file listed the models using Unix filenames. If needed, the template now converts the .LST filenames back to the operating system the template is running on.

Undo

- **Case 26910**

Undo for Detach was not working. Now fixed.

- **Case 26816**

When trying to UNDO something related to a set in long format, the UNDO sometimes could not be applied. This has been corrected.

- **Case 26684**

The 'Node Drag' panel presents users with a model selection menu when multiple models are present (and one hasn't been chosen). However, the Undo button is available in the background at that point, which when clicked on resulted in a crash. This has been fixed by hiding the Undo button till a suitable model has been chosen or if a single model is present.

Units

- **Case 27833**

The unit conversion of the FAC field on *MAT_FABRIC could be incorrect for some options. This has been corrected.

- **Case 26882**

ICFD_MAT units conversion logic has been added.

Write

- **Case 26528**

*CONSTRAINED_COORDINATE was written out incorrectly as *CONSTRAINED_BUTT_WELD. This has been corrected.

- **Case 26383**

*INCLUDE_ lines were not being written out to master models if the Alphabetical output style was chosen. Now fixed.

Ztf

- **Case 28006**

PRIMER could crash when writing a ZTF file for a model containing latent parts referenced by beams. This has been fixed.

2.2 D3PLOT

2.2.1 Bugs Fixed in 12.0

Contour

- **Case 24489**

The criterion (direction) plot of 2D principal stress was being displayed in the wrong direction for a uniaxial shell test model in which the stress was exactly aligned with the element local axes.

The problem was a wrongly written ill-conditioning check which would only be triggered by this exact set of circumstances, ie pure compression with exactly no shear.

The calculation has been fixed and will now deal with that condition correctly.

Cut Sections

- **Cases 12634 and 24276**

When multiple models exist in a window and a cut section is active the user would expect that screen dragging the section would result in the sections cut through each model remaining co-planar (in deformed space).

This was true for drag translation, but for drag rotation the sections could sometimes end up no longer being co-planar. This was due to ill-conditioning in the calculation that turned a cursor drag motion into a rotation which could give subtly different results for the different models in the window, and build-up of these differences would quickly result in different plane orientations.

Drag rotation of a cut plane through multiple models in a window now imposes exactly the same rotation on the section in all models, solving this problem.

Data Access

- **Case 24649**

The data component 'Strain Rate' now appears with the rest of the strain based data components. For backwards compatibility it also still appears under the miscellaneous data components but it has been made more obvious that it applies to thin shells as well as solids and thick shells.

Envelope

- **Case 22370**

Historically an attempt to 'capture' in REPORTER a D3PLOT image that shows an envelope plot has failed. Either it gave no results or - worse - results that were wrong.

There were two reasons for this:

(1) The actions required to initiate an envelope plot in D3PLOT during playback of the REPORTER template were missing, meaning that the plot was not set up correctly.

(2) It was also possible to select a subset of states during 'Capture' of the image, but this list of states was not stored and so was lost during playback.

These two problems have been fixed as follows:

(1) The necessary actions to initiate an Envelope plot now take place, and it is set up correctly.

(2) In the REPORTER playback case this plot will always span all states in the model.

This 2nd point, using all states, is what most users do most of the time, so it should be satisfactory. However if users try to define a subset of states in the REPORTER 'capture' phase this will be intercepted and changes to all states, with a warning message giving an explanation.

Graphics

- **Case 26107**

Capturing a screen image in REPORTER from D3PLOT when running in 'batch mode' (which means no screen window is shown) resulted in failure on Linux and a crash of D3PLOT.

The crash occurred because D3PLOT was attempting to use hardware accelerated graphics when not running in a physical window, ie when the graphics card was not not properly 'connected to' the display. It has been fixed by reverting to software shading when in batch mode.

- **Case 25581**

When Shift Deformed was used in conjunction with Magnify Displacements the wireframe overlay on the plot could sometimes 'lag behind' when shift deformed was turned on/off, and not be drawn in the correct position until an explicit redraw command was issued.

- **Cases 25581 and 21652**

When an image is captured containing Airbag Particles, SPH elements or DES elements the elements are now automatically drawn as spheres in the image if the on screen symbol has been set to point.

This change has been made as although 'points' are quicker to draw they don't always appear the correct size in captured images due to limitations in the maximum point size supported on some machines when drawing to an image.

Using spheres is slower (this doesn't matter as they are only used when capturing the image) but the symbols end up the correct size.

- **Case 25428**

Predictive picking of Spring and Seatbelt Parts in D3PLOT would incorrectly put the label at the coordinate 0,0,0.

Version 12 has been modified so if the user is picking PARTS and the element selected is either a spring or seatbelt the label is now drawn at the element centre.

- **Case 25013**

When capturing an image (static or movie) any restraint symbols, ie SPCs, would fail to appear in the image file despite being visible on the screen.

- **Case 24913**

If you used the up and down arrow keys to cycle between models and then changed the drawing mode from SH to LI, then all the models changed to LI only if M1 was drawn. If a different model was being drawn the drawing mode got reset the next time you pressed up or down to cycle to another model. This has been fixed.

- **Case 24526**

In version 11 the Overlay mode for SPH and ABP elements could be set through the 'Display Options' menu but the Quick-Pick Overlay option didn't work for these element types.

- **Case 24462**

If the mesh overlay was changed from 'Free Edges' to 'All Edges', the overlay updated on all the solid, beam, shell and thick shell elements but not on SPH elements.

- **Case 24284**

If velocity arrows were drawn on SPH elements they would not be seen if the length of the velocity arrow was less than the radius of the SPH element. In version 12 the velocity arrows are now drawn with their tail positioned at the surface of the SPH element instead of the element centre so they are not lost inside the element.

- **Case 24282**

If the option to re-read models was used when multiple models had been loaded into D3PLOT then any window based properties such as SPH radius, beam plotting mode etc which had been set would not be applied correctly to the model after the re-read operation had completed.

- **Case 23952**

When using software computed normals (not the default on modern machines), the lighting in a model in a second window could occasionally go wrong being both 'the wrong shade' and also often too bright.

- **Case 23877**

If data values were displayed using dynamic labelling on spotweld solids then the data value would be displayed on all the visible faces which made it very difficult to read unless you zoomed right in on each spotweld.

Version 12 had been changed so that the data value is now displayed on just one face of the spotweld solids.

- **Case 23357**

Part-based elements would sometimes fail to draw in a state if the following sequence of operations was followed:

- Read model, turn off 'entity' switch for some category, eg solid.
- Close model.
- Reopen model, or open a new model with similar content.

At this stage no solids will be drawn because the entity switch for them is off.

Turn the entity switch on and the solids fail to appear in that state, although they will appear when the next state is rendered.

The problem was that part-based elements such as solids are subject to a 'coarse clipping' test which means that they will only be drawn if the part which contains them is on the screen. This test is based on the bounding box round the part, and this was being set up wrongly if the entity switch for that part's element type was turned off.

The same problem of 'started new model with entity switch off' could also be provoked in other ways, giving similar problems.

This has been fixed by making sure that when a part is first rendered within a state there is an 'is this the first time I have drawn you?' test which calculates the bounding box properly.

- **Case 23022**

If a ZTF file is present D3PLOT can obtain restraints and constraints on nodes, and display these if the appropriate switches are turned on in the Entity panel.

However this has caused some confusion since the symbols used for both REStraint and CONstraint are the same, making it difficult to tell one from the other if more than one category is turned on.

To compound the problem one class of constraint is 'Node on a rigid part', and if this is selected then all degrees of freedom are shown as constrained (which is correct), but this can be confused with the actual restraints - if any - on the rigid part.

So an extra category 'Restraint on rigid part' has been added which will show restraints in the appropriate degrees of freedom on nodes on rigid parts which have been restrained on their *MAT card.

Groups

- **Case 25953**

If D3PLOT read a GROUP .grp file written from an older version of D3PLOT and then you used the blank menu to blank groups, D3PLOT could crash.

Image

- **Case 21985**

If an image was captured containing multiple windows with envelope plotting turned on then the first image captured for a data component could have blank images where some or all of the windows should have been. If a 2nd image was captured without changing component then it contained all the windows.

Labels

- **Case 25926**

In the entity panel the 'Attached Nodes' switch for displaying node names did not work.

Measure

- **Case 25854**

Deleting the current measurement was not removing it from the graphics window. Pressing SH or moving the model updated the display. It now updates straight away.

The same problem has been fixed with the Delete All button.

- **Case 25023**

Measure node-to-node would return angles as NaN if the nodes were coincident. They are now returned as 0.0.

Menus

- **Case 25949**

When using the D3PLOT->T/HIS link the position of T/HIS graphs would get reset if the user minimised and maximised the main application window or iconised it.

In a standalone D3PLOT or T/HIS session manually resizing the application window by dragging out it's corners would also reset the position of any graphics windows that the user had manually moved but not any that he had manually resized.

- **Case 25942**

If the mouse scroll wheel was used while the mouse was either over the border of a window or over the green bar that divides the graphics area from the right hand menus the program could stop responding.

- **Case 22300**

If the option to see the size of the graphics windows was used (UTILITIES->GRAPHICS) then the new sizes were not applied to windows that were not on the current page even if the window tabs were selected for those windows.

- **Case 14239**

If D3PLOT had multiple windows open and the Wn selector had been used in one of the graphics windows to deselect that window in all the menu panels then the window tabs in the graphics windows would get out of sync with the ones on the menu panels if the user created a new window.

Movies

- **Case 25414**

If a movie was captured when you had a movie or static image in the background of the D3PLOT window and the T/HIS link was also being used then the model in the movie would appear to jump from one location to another after a few frames. This wouldn't happen on screen - just in the captured movie. This has been fixed in version 12.1.

- **Case 24668**

If a background movie was loaded in a Window in D3PLOT and a settings file was written out the movie would be loaded correctly when the settings file was replayed in a new D3PLOT session but the display of the movie background was sometimes not turned on.

If a number of points had been selected to align the model with the movie the points were not written out to the settings file and would be lost when the settings file was reloaded. In version 12 the point details are now stored in the settings file and can be edited after reloading.

Multiple Models

- **Case 24687**

If a reference node was setup in model 1 before a 2nd model was read in the reference node would not be applied correctly when values were contoured. Instead of displaying the relative values the absolute values would be displayed. Turning the reference node option off and on again would fix this problem.

- **Case 24675**

If a second model was read into Window 1 in D3PLOT then sometimes the model would be added correctly to the window but it's visibility switch (Window->Edit Windows->W1) would not be set correctly. This would often happen if the 2nd model contained a properties file.

Part Tree

- **Case 25011**

If the option to list contact surfaces in the part tree was turned on then two times as many rows would be displayed as there were contact surfaces in the model. These extra rows would be labelled with '0' and you couldn't do anything with them, but they shouldn't have been listed.

Properties

- **Case 24944**

If the option to reload a properties file into D3PLOT was used then after selecting a properties file for a model the tick box option to read the properties file wasn't automatically selected. This meant that if after selecting the file the user just pressed 'Apply' the file wasn't actually read.

Quick Pick

- **Case 25065**

If the Quick-Pick Find option was used while a model was loaded into different windows the cross-hairs highlighting the item were only drawn in the 1st window.

- **Case 25052**

If the Quick-Pick Find option was used while multiple models were loaded into different windows the location of the cross-hairs highlighting the item didn't update correctly if the magnification factor for displacements was changed. The cross-hair was located correctly in any windows containing the 1st model but not in windows for other models.

Read

- **Case 25876**

If multiple models were read into D3PLOT and D3PLOT was unable to read the LSDA/binout files for one of the models because it was still running then D3PLOT could crash if the user deleted all the models in D3PLOT.

- **Case 25092**

D3PLOT V11 will only read individual files up to 2GBytes in size (single precision).

D3PLOT V12 will read up to 8GBytes (single precision) or 16GBytes (double precision), these limits being determined by internal addressing issues.

These limits will be removed in future releases.

- **Case 24461**

Case 24461If a model was opened and the factor used to scale the radius of SPH elements was modified the new factor wasn't applied to a second model when it was opened.

Scripting

- **Case 25389**

The maximum number of widgets that could be made in a window was limited to 1000. This can now be changed by using Options.max_widgets.

- **Case 25064**

The Blank/Unblank, IsBlanked and IsDeleted functions crashed D3PLOT if you tried to use them with nodes.

- **Case 24256**

The following 2 bugs in the JavaScript function GetData() function have been fixed:

  - It always returned zero for segment contact forces.
  - It gave an error saying that the SURF item type was not valid.

An extra optional argument has been added to the GetData() function to switch whether blanking should be considered or not when calculating results (default is not to consider blanking). This is relevant to nodal contact forces as by default this function will return the sum of all forces at a given node for all surfaces using that node. Users can now blank all but the contact surface(s) of interest and turn this option on to restrict the results to the contact surface(s) they want.

Two new functions have been added to help working with contact surfaces.

GetSegmsInSurface() returns the start and end indices of the slave and master segments in a given contact. This can be used to loop over the segments.

SpoolNodesInSurface() can be used to spool through the nodes on the slave and master sides of a contact.

Settings File

- **Case 24676**

If a settings file was written out while the D3PLOT->T/HIS link was active the model index written to the settings file for any D3PLOT windows that were created after the T/HIS link had been started were wrong. This could cause a crash when the settings file was reloaded in another session.

T/HIS Link

- **Case 25306**

When the D3PLOT->T/HIS link was used with a model containing X-Sections a local triad was drawn on the X-Sections by default using a length of 1.0 for the triad axis as the correct length wasn't being calculated. This didn't matter if your model was in mm but if the model dimensions were in m then drawing the triad with a length of 1.0 resulted in a huge triad for each X-section.

D3PLOT 12.1 now only draws a triad if the user has turned on the option in the 'Display Options->X Section Symbols' menu and the triad size is calculated correctly.

- **24537**

When using the 'Locate' option in the D3PLOT->T/HIS link it wasn't obvious to users how to remove the locate symbols or to clear the items they had sketched. The 'Locate' popup menu now contains a 'Clear All' option that removed any sketched items or locate symbols.

User Interface

- **Case 22773**

In version 12 the maximum length for filenames (including the directory) has been increased from 256 to 512 characters

NOTE : Internally Windows supports filenames of up to 32768 but currently it does not support pathnames longer than 260 characters within Windows Explorer or the command line interface..

User Defined Data Components

- **Case 24269**

D3PLOT 11 could crash if user defined data components were created in a model contains SPH elements.

X-sections

- **Case 26083**

Selecting a LoadPath component could crash D3PLOT if the d3plot.components file had not been read or did not contain data to set up X-Section components. These are now made unavailable for selection if this is the case.

2.2.2 Bugs Fixed in 12.1

Adaptivity

- **Case 27447**

D3PLOT crashed during model open of an adaptively remeshed analysis using Element Free Galerkin (EFG).

This was because it got confused about which of the adaptive remesh families it was currently in during the special actions taken at model open, and this EFG analysis exposed that problem.

Fixed by removing this 'which family' confusion.

Ale or euler

- **Case 27836**

Version 12 of D3PLOT could sometimes generate an incorrect warning message regarding the number of ALE materials in a model even though the model did not contain any ALE materials. Version 12.1 has been modified to suppress this warning message when it is not appropriate.

Animation

- **Case 26620**

If an animation was set only be displayed once 'Anim > REPEAT > ONCE ONLY' then at the end of the animation the state slider would show the last state but D3PLOT would think that the current state was the 1st one. if the screen refreshed the 1st state would then be displayed.

- **Case 23438**

Reading a background movie from a series of static images was not working.

Blob plots

Case 26531

Arbitrary text on 'blob plots' (Utilities->External data) was not being displayed.

Checking

- **Case 27029**

In 3d meshes D3PLOT attempts to find the facing element for each face on a solid or thick shell, since this is required for blanking and other calculations. To do this it has to consider the topology of these elements, and therefore it detects various 'cross faces' situations, typically symptoms of gaps in the mesh. If the user turns on the D3PLOT_LIST_CROSSED environment variable these are listed to the controlling terminal when the model is read.

A slightly grey area arises where two triangular faces, for example on tetrahedron or wedge elements, are used to match the four noded quad face on a hexahedron. This isn't strictly wrong, since all nodes have mesh continuity, but it is not good practice either.

By default D3PLOT accepts this partial face match silently, but if the D3PLOT_LIST_CROSSED environment variable was set it would list these elements as having crossed faces, making it look as if the mesh had genuine errors - rather than just less than optimal quality.

This has been changed so that by default these situations are no longer listed as 'crossed'.

However if the acceptance of this meshing method is turned off by setting the environment variable D3PLOT_PERMIT_CROSSED to FALSE then these elements will still be listed.

Contour

- **Case 27423**

Setting the max and min contour levels were only getting applied to the first model if there were was more than one model in a window.

- **Case 27044**

The option to set the Max + Min contour levels was not working correctly if more than one window was open and different values were set for each window.

D3PLOT used the envelope of the values set in all the windows, so they all ended up with the same contour levels.

- **Case 26863**

D3PLOT could crash if after doing an ISO surface plot of an ALE data component you then swapped back to a structural data component.

- **Case 26460**

If multiple models were displayed in the same window then user defined min/max contour values were not applied correctly to all the models (they were only set on the 1st model). If the up and down arrows were then used to toggle on/off the models visible in the window the old wrong contour bands were displayed for any models apart from the 1st one.

- **Case 26424**

If multiple models were displayed in the same window then user defined min/max contour values were not applied correctly to all the models (they were only set on the 1st model). If the

up and down arrows were then used to toggle on/off the models visible in the window the old wrong contour bands were displayed for any models apart from the 1st one.

- **Case 26285**

When the following sequence was performed:

- Read new model
- LC plot for a line contour plot of data
- Animate
- While still animating switch to SI plot for shaded contours

Then sometimes all shaded image plots in states 2 to n-1 could show no contours.

This was caused by the fact that an LC animation was sometimes not setting up enough 'remembered' information for the subsequent SI plot to 'know that' there was data to show in those intermediate states.

A workaround would be to perform the SI plot first, or to set the contour scanning mode to 'all states'.

Now fixed: data is remembered correctly regardless of the order in which plotting modes are used.

Cut section

- **Case 27779**

If you did a contour plot and excluded an element type from a cut-section, the values for the contour bar and max/min values were based on what would be visible if nothing was excluded rather than what is actually visible.

- **Case 27296**

The 'exclude' option in cut-sections was not working for parts that straddled the cut-section plane if the cutting plane was swapped from deformed space to basic space.

- **Case 27133**

The documentation and calculation of cut-section moments in shells wrongly stated that the moment resultant Mxy generated a torsional moment about the shell local Z axis.

In fact Mxy generates (warping) torsion about shell local X and Y axes, and the code has been corrected to show this. However testing reveals that the Mxy resultant moments as reported in shells are a poor guide to warping torsion, so these numbers should not be relied on.

- **Case 26728**

Cut section forces in shells were not being computed if the user had used the *DATABASE_EXTENT_BINARRY to turn off the output of both shell forces and moments (RLTFLG) and also thicknesses (ENGFLG).

If a ZTF file was present it *should* have used the 'raw' thickness in that, taken from the *SECTION_SHELL card, but it was failing to do so, and was also failing to use the shell mid-surface stress tensor correctly.

Both problems now fixed.

- **Case 26608**

It was not possible to select Cut Section data components in the XY Data and Write menus for models read from a Nastran OP2 file.

- **Case 26171**

If a CT plot with a cut section was plotted for a volume 3 (ICFD) analysis the results on the cut section was drawn in black unless an SI plot had previously been done.

Data components

- **Case 27669**

If a volume 3 (ICFD, CESE, EMAG) vector data component was selected then the correct data component was plotted for SI, CT, LC, CL plots but for VEC plots the default Vol 3 vector data component was always plotted not the one that had been selected.

- **Case 27544**

D3PLOT could plot the wrong data values for contact data components if a model was read in with a CTF file which had some of the default data components removed from it using *DATABASE_EXTENT_INTFOR. D3PLOT 12.0 did not know about the *DATABASE_EXTENT_INTFOR card so it assumed that any data written to the CTF file was in the standard format. IF some components were removed it would therefore end up reading data from the wrong locations in the CTF file.

- **Case 27543**

If you read a CTF file into D3PLOT which had all the segment based data components excluded from it (using the *DATABASE_EXTENT_INTFOR option) then D3PLOT would crash when you tried to plot any of the contact data components. D3PLOT now detects that the file doesn't contain any data and prints a message to the dialogue that the CTF file will be ignored.

- **Case 27542**

If 'Contact Gap' was chosen as the data component under the Contact component category then the plot would not show the correct values as the contour bar limits were not set correctly.

- **Case 27438**

If a vector data component was selected for a CESE analysis then the vector arrows were drawn showing an incorrect magnitude and direction if D3PLOT was using the default 'vector' graphics mode. All other plotting modes (CT, SI, CL etc) would show the correct magnitudes.

If the graphics mode was swapped from 'vector' to 'direct' then the arrows would be drawn with the correct magnitude and direction.

- **Case 27435**

If a vector data component was selected for an ICFD analysis then the vector arrows were drawn using the correct magnitude and direction but at the wrong node if D3PLOT was using the default 'vector' graphics mode. All other plotting modes (CT, SI, CL etc) would show the correct magnitudes.

If the graphics mode was swapped from 'vector' to 'direct' then the arrows would be drawn at the correct node.

- **Case 27234**

In a 2D ALE analysis D3PLOT reported zero for all the ALE components.

A workaround in 12.0 is to plot a data component like X-Stress 1st and select 'Layer 1' as the surface and then swap back to the ALE data component.

- **Case 26619**

In version 12.0 the data component 'Level Set' was not available in ICFD analyses. This has been fixed in 12.1 by updating the 'multiphysics.components' file.

Deform

- **Case 26375**

If you plotted a user defined data component that used displacements, velocities or accelerations, and then turned on reference node to plot it relative to some nodes, the contoured values did not update.

This is because D3PLOT caches data to speed up plotting, but in this case it should have

recalculated it as it was out of date.

A warning message is now issued if the model contains user defined data components and the reference node settings have changed. The user can then select if they want to recalculate the values (and will not be asked again in the session).

Envelope

- **Case 27213**

If the state selector for the 'Envelope' option was open when a model was deleted in D3PLOT then the state selector would remain displayed even though the model had been deleted. If the user then clicked on one of states in the envelope menu D3PLOT would crash.

Femzip

- **Case 27216**

When reading FEMZIP files D3PLOT fires off a separate reader process for each model opened, and then the main D3PLOT and reader processes 'talk to' one another as they extract data.

In case the child reader process locks up or crashes the main D3PLOT process has a timeout limit after which it will assume that the reader has died, and by default this is set to 60s.

This is usually adequate, however when opening a very large FEMZIP dataset, typically more than about 6GBytes in total, the time taken to 'scan states' can legitimately exceed this timeout limit, leading the D3PLOT process to think that the FEMZIP reader has died, when in fact it is still busy.

To fix this D3PLOT now scans the FEMZIP dataset and adjusts this timeout limit according to its size.

(The timeout limit can also be set manually via the environment variable D3PLOT_FR_TIMEOUT.)

General

- **Case 26722**

Using a 3D mouse (“Spaceball mouse”) in D3Plot in window 2 or above could give incorrect model translations. This has been corrected.

- **Case 26085**

Problems could occur on Linux platforms when D3PLOT (or PRIMER or T/HIS) are run from the command line without having set up any of the environment variables that they expect. For example normal usage will set up OA_INSTALL to define where the executables are installed, but a free-standing installation that is run manually may not have this defined.

The problem would be most likely to arise if relative filename syntax was used to run the codes, for example '../executables/d3plot12.exe'.

Problems encountered might be:

- Not picking up any oa_pref file in the installation directory.
- Not picking up d3plot.components or multi-physics components files.
- Not finding the ffmpeg run-time libraries required to decode some image or movie formats.
- Not finding manuals

.. and so on.

The problem was due to a failure to extract the original directory from which the executable was run if this was not defined explicitly.

This has now been fixed, and Linux executables now 'know' the directory from which they are being run (in the same way that Windows executables know this), even if relative pathname syntax is used.

Graphics

- **Case 27623**

D3PLOT 12.0 could crash when animating data from a volume 3 (ICFD, CESE, EMAG) analysis if the animation contained a plot state that the user had not already looked at. The crash wouldn't always occur as it was caused by a random memory overwrite.

- **Case 27588**

If a model contains solids and shells, and the following sequence of operations is performed:

- Set Entity switch to solids only
- Autoscale
- Model reread

Then at this stage, correctly, only solid elements in the reread model will be shown. However if the user now turns on the entity switch for shells these may not be drawn until the scale is changed by a considerable amount.

The error was that the bounding box around the shell parts was not being calculated correctly when their visibility was turned back on again, meaning that they could be 'clipped' from the view.

Now fixed by making sure that when the visibility switch for any item is turned on for the first time then its bounding box is calculated correctly.

- **Case 27464**

If the entity panel was used to the on the display of contacts then the individual options for contact segments and contact nodes didn't work as expected. If you turned on the option for segments then both segments and nodes were displayed, while turning on just the contact nodes only displayed the nodes as expected.

In version 12.1 turning on the option for contact segments now only displays the segments.

- **Case 27441**

If you read in 3 models into one window and then removed model 2 using Window->Edit, pressing the up or down arrow key would try and display model 2 and show a blank screen, i.e. it was looping in the sequence 1->2->3->1->2->3 when it should have been 1->3->1->3.

Pressing the up/down arrow keys now skip any models that have been removed from a window.

- **Case 27440**

If you used the '=>' button on the Window->Edit menu to put all models into Window 1 and then read in another model, it's default display mode (in Window 1) got set to Wireframe when it should have been shaded and it's colour got set to black.

- **Case 26903**

Testing revealed various problems with Volume III (ICFD, CESE, EMAG, etc) data which have been fixed:

(Rarely) a crash when contouring interpolated Vol III data due to a failure to initialise internal data properly.

Poor quality ISO surface plots because the data averaging method used for Vol III data was the same as for structural, ie 'Medium resolution', which is not averaging at all. Now swapped by default to low resolution.

And some other niggles that could affect images.

All fixed.

- **Case 26650**

D3PLOT crashed when displaying an SI plot that contained beams with data and beams were being displayed in a mixture of 'current' and 'shaded' modes.

It got into a muddle internally about lighting. Now fixed.

- **Case 26338**

If spotweld clusters were displayed in D3PLOT using the option to display them as spheres then the location of the sphere wasn't always calculated correctly.

Groups

- **Case 27791**

Some parts were missing when writing to a group ASCII file. This could happen if the part labels were not in ascending order in the model. Now fixed.

- **Case 27378**

The labels for X-Sections or Load Path Segments were written as 'undefined' in group ascii files, which meant they could not be read back in. They are now written with the correct labels of 'XSEC' and 'LOAD_SEGS'.

Javascript api

- **Case 27244**

If the JavaScript debugger was running the window could obscure the D3PLOT selection menu when the 'Select' function. From version 12.1 onwards the JavaScript debugger window is automatically hidden at the start or any pick/select operations so that it doesn't obscure any menus and then re-appears at the end of the picking/selection operation.

If the debugger windows has been moved out of the main application window and onto the desktop then it remains visible.

- **Case 26904**

The Javascript API function GetElemsAtNode() takes the arguments [Node id], [Element type], where [node id] is an internal node index if +ve, and an external node label if -ve.

This function was wrongly checking the node index (+ve case) against the total number of [Element type] in the model, so this function could fail if node ***I*** was given and the total number of elements of that specified type was less than ***I***, otherwise it would be fine.

If a -ve node label was given it would look up the equivalent index of the element of this label, giving the wrong node index.

Both errors were due to using the wrong type code, that of the element, rather than 'NODE' when performing these checks / lookup, now fixed.

- **Case 26879**

Step and Next did not work in the JavaScript debugger if the script file did not have a carriage return on the last line. Now fixed.

- **Case 26817**

The JavaScript debug window disappeared when using CreateWindow().

Local axes

- **Case 26479**

Selecting a local coordinate system in the measure menu would sometimes crash D3PLOT.

Measure

- **Case 26932**

It was not obvious how to resume measurements if the user pressed 'q' for quick pick mode.

When 'q' is pressed, the measure panel will now only show a 'Resume Measurement' button which should be pressed once the user has finished their quick pick operations.

- **Case 26387**

If measurements were created without labels and saved to a settings file, when you read the model in again the measurement labels were set to '*BLOCK'. They are now set to be blank.

Menus

- **Case 27047**

Pressing the button to make the principal plot arrows the same length did not have any effect.

- **Case 26753**

If you had multiple models loaded into D3PLOT and the 1st model didn't have any Airbags with Gas Properties then clicking on the 'Gas Properties' button didn't do anything (it should have displayed a window with Model tabs that allowed you to set the Gas Properties in other models).

The 'Gas Properties' and 'Airbag Chambers' menus were also not updating properly when the model tabs were changed if they had been displayed via the 'Data Components' menu (they did update correctly if displayed via the 'Display Options').

- **Case 26355**

In version 12 the popup menu for selecting EXTRA data components didn't always show all the available components if multiple models were loaded with different numbers of EXTRA data components.

Movies

- **Case 26546**

Background movie playback could be 'jerky' if the movie and analysis was set to something other than every state.

If you used the state slider in the bottom right hand corner and selected a state that wasn't in the animation sequence the movie could reset to the first frame. It now stays on the previously displayed frame.

Nastran

- **Case 27790**

D3PLOT failed to read in a NASTRAN OP2 file that contained a grounded spring. This has been fixed.

Preferences

- **Case 27229**

A spurious error message about the preference

`d3plot*gtune_vbo_limit` being set to -1

could appear if the TUNE panel option to save settings was used.

This was harmless, and would have no effect, but could worry people.

Now fixed.

Properties

- **Case 27203**

Clicking on the Load Path button in the Properties menu caused D3PLOT to crash.

Quick pick

- **Case 26390**

When Quick Pick by include file (or entity) is used in a large model to blank a screen area the time taken to perform the blanking operating is quick, but a subsequent 'undo' using the middle mouse button can be very slow.

This is because of the way the 'undo' information is stored, resulting in some lengthy processing when it is used to restore the graphics to their former state.

This processing has been speeded up and should be at least 10x faster, giving an acceptable speed.

Read

- **Case 26747**

If the default naming convention for d3plot/PTF files generated by LS-DYNA was changed so that they were named 'd3plot.ls', 'd3plot.ls01', 'd3plot.ls02' ... then D3PLOT would fail to open the file and would report it couldn't find a database file called 'd3plot'. This has been fixed in version 12.1.

A new preference option has also been added that can be used to change the default file extension used in the file filter when searching for d3plot files.

```
d3plot*file_filter: *.ls
```

would change the default file filter for d3plot files to '*.ls'

- **Case 26414**

LS-Prepost .db files were not reading into D3PLOT.

- **Case 26352**

When performing a (File) Model Reread operation D3PLOT would close the existing model, reread it from disk, and then restore the model in its new form in all windows in which it previously was present.

As part of this process it had to 'remember' the various per-window settings used for the model, and this process was carried a bit too far because, among other things, it remembered the final animation frame number of the old model. This meant that if the new model had a different number of states then things could go wrong if the user chose to go to the last frame using the **>>|** button.

Now fixed: the save/remember process during a model reread now only remembered non-model-specific information for the windows involved.

- **Case 26127**

If the option to 'Auto Open' is turned off in the model read menu in D3PLOT then the user can specify an alternative ZTF file for D3PLOT to read instead of letting D3PLOT search for a matching ZTF file in the same directory as the PTF file.

This all worked for the 1st model but the 2nd time onwards when you use the Read Menu the user defined ZTF was ignored and the default one found in the model directory was always read.

Sections

- **Case 26729**

In a model containing database cross-sections we have historically computed the space required to store centroid and area data for all states, assigned this, and then populated it either from the LSDA file or, if that is not available, from calculated geometry.

However if the model is re-scanned for any reason and finds more states, as can happen when the job is still running, the length of the space set aside for these data is not recomputed, with the result that when cross-section data from later states is accessed a crash can occur due to memory corruption, probably in somewhere completely unrelated.

Fixed by changing this to assign storage on a per-state basis on demand.

Selection

- **Case 26920**

A crash could occur (rarely) when predictive picking by part was used in a model with:

+ Springs and seatbelt elements.
+ The ls-dyna flag that created beams in the output file on top of these elements.
+ A ZTF file in use.

This could cause internal confusion in D3PLOT because the beams created by LS-DYNA have the same part id and nodal topology as the springs, leading to confusion about what that particular PART actually was.

The special logic in Predictive picking that highlights items under the cursor could get confused by this

T/his link

- **Case 27783**

D3PLOT could crash when using the D3PLOT->T/HIS link if the Locate->Display (Only) option was used in T/HIS to locate an item in the model in the D3PLOT window.

- **Case 27149**

If you changed window layout while using the D3PLOT->T/HIS link the T/HIS graphs could be incorrectly positioned and end up overlapping the D3PLOT windows.

- **Case 26463**

If all the models in D3PLOT were closed while using the D3PLOT->T/HIS link and the Open Model menu was then displayed in D3PLOT to select a new model an incorrect error message similar to

'Cannot find database file /THF G_RESIZE 1 1 0.00000 0.574767 0.21835 0.925103'

could be displayed.

User defined

- **Case 27662**

When user defined binary data components are created in scripting they save their results to '.ubd' files, and by default these are written to the same directory as the results from LS-DYNA. When a model is read, either in this session or a new D3PLOT session, then this directory is scanned and any .ubd files found are read and their user-defined data component names become available for use in the current session.

This all works well in the normal situation of the job directory permitting write access, but some users run their analyses on remote machines which write results to directories that are read-only. In this situation the .ubd files cannot be written there, and the following alternative logic is used:

(1) If a 'location' argument is given to the CreateUbinComponent() Javascript function call then this is used instead.

or if that is not defined

(2) If the preference **`d3plot*ubd_file_location`** is defined then that location is used instead

or if that is not defined

(3) The files are written in a temporary directory, which can be defined on a per-used basis via the OA_TEMP preference or by using preference **`d3plot*temp_dir`**. If these are not defined then the user's home directory or the system default temporary directory are used instead.

This all works for writing files, but causes problems when a new model is opened or a new D3PLOT session is started since it did not search for files in anywhere other than the directory containing the normal results files.

This has been fixed: if the preference **`d3plot*ubd_file_location`** is defined then it will search the job directory as before, and if no .ubd files are found there it will also search the directory defined in the preference.

In addition the syntax for this preference, and also the 'location' argument in CreateUbinComponent() has been extended to permit the special word JOBDIR to be used. This means 'the normal results directory' which, in itself, is not very helpful. However you can append

further relative pathnames to this, for example:

JOBDIR/.. means the directory above JOBDIR
JOBDIR/../../my_ubd_files means two directories above, and then down into directory my_ubd_files.

In this way if your job directory itself is read only, but a directory above that is writeable, you can specify that your .ubd files are placed there instead.

- **Case 27191**

If you created a user defined component that read data from a text or javascript file and pressed 'Reread file', the name and data source of the previously defined component could get overwritten. This has been fixed.

- **Case 26483**

Two problems with User Defined Binary Data (UBIN) components have been fixed:

(1) When using the default settings D3PLOT will save UBIN data in a filename analysis_0001.ubd.

The internal logic that creates that name assumes that name attempts to strip any extension off the original filename, looking backwards from the end of the filename for a '.' If the original name did not have an extension, typically filename 'd3plot', then it wrongly continues to search backwards up the pathname string. This works OK unless that pathname itself contains a '.' character, in which case it wrongly treats everything after that character as being the extension, strips it off, and puts the .ubd file in teh wrong place.

For example if the analysis was in H:\my.files\d3plot the UBD file would end up being called H:\my_0001.ubd.

(2) One workaround for this is to define the LOCATION argument to the CreateUbinData() call, specifying an explicit location for the UBD file.

This works, and does not suffer from the filename parsing problem above, but it has the unfortunate effect of overwriting the component name with this pathname. This is due to an internal error handling the function arguments.

Both problems have been fixed.

Utilities

- **Case 27798**

The file written out from the PTFCUT menu that saves the current selections commented out the last line '$ APPLY', so that when it was read back in it didn't automatically write out the cut down model. This was intentional.

However, this proved confusing if it was to be used with the **`-ptfcut`** command line option as it had to be uncommented before it would work.

This line is now written out uncommented and is ignored when the file is read in from the ptfcut menu.

- **Case 27188**

In the Utilities->PTFCUT, if you tried to write out a 'Reordered' database with *DATABASE_CROSS_SECTION data, it could fail on Unix machines.

- **Case 26493**

The Delete All button in the target markers menu didn't work if you had multiple models in D3PLOT.

- **Case 26492**

Target marker sizes were not being carried over from PRIMER into D3PLOT.

Volume clip

- **Case 27668**

There were no dialogue commands to define the centre of a clipping volume using a node. This can now be done with

/CREATE NODE_CARTESIAN or NODE_CYLINDRICAL or NODE_SPHERICAL

Write

- **Case 26872**

If the 'Coincident Element' option was used in the WRITE menu D3PLOT would crash if the model contained any of the following

Spotwelds
X Sections
Load Paths

Ztf

- **Case 26393**

Models with MIG weld line data could crash D3PLOT. Turning off reading the ZTF file would prevent the crash.

2.3 T/HIS

2.3.1 Bugs Fixed in 12.0

Curve Table

- **Case 25839**

If the curve table was set to 'Group Common Items' then the Style column was correctly shown for each row using the description 'Mixed'. The component, model and directory columns however could show incorrect descriptions.

- **Case 25838**

If columns were turned off in the curve table using the curve table view menu then the curve table wasn't updated correctly until the view menu was dismissed.

- **Case 24976**

If the display of columns in the curve table were turned on or off the positions of the buttons and scroll bars used to select which curves were displayed on each graph did not update immediately.

If the curve table was resized or dismissed and then re-displayed then the positions updated correctly.

Filtering

- **Case 25542**

If the auto regularisation switch was on any operations that used it would always take the value specified to regularise the curve. This was wrong if the curve had a smaller timestep than the one specified as you would effectively be filtering the curve. Any operations that use the auto regularisation now use the smaller of the two values.

The curve label now also includes the regularisation timestep used, e.g. if you use the C60 filter with regularisation on the label is now 'Node 1 (Reg 0.0001) (C60)', rather than 'Node 1 (C60)'.

General

- **Case 21840**

In version 11 of T/HIS it was not possible to turn off the display of individual injury criteria if more than one injury criteria had been calculated for the same curve.

Version 12 has been modified so that the curve properties menu now allows the display of each injury criteria to be turned on and off separately in both the graph area and the legend.

Groups

- **Case 24257**

If T/HIS read in an ASCII file containing PART GROUP definitions the groups would not be created correctly if they were not defined in order of increasing group ID in the file.

In version 12 the PART GROUPS can be defined in any order.

Menus

- **Case 24482**

In version 11 of T/HIS scrolling the list of curves in the curve manager using the mouse scroll wheel was very slow on LINUX if shaded buttons were being used. Turning off shaded buttons by setting the button graduation to zero (Top Tool Bar : Options->Menu Attributes) fixed the problem.

In version 12 the scrolling speed has been improved but it is still slower on LINUX if shaded buttons are used compared to flat buttons.

Operations

- **Case 23693**

If the automatic regularisation option was used with the FFT or IFFT function and the time interval was very small compared to the length of the curve then the FFT function could take a long time to complete. T/HIS now checks to see if the STOP button has been pressed within the FFT function and aborts the curve operation if it has.

Quick Pick

- **Case 22292**

When using the D3PLOT->T/HIS link the T/HIS Quick Pick option didn't always work correctly after doing something where the user could pick curves interactively (Operate Menu etc). After using the Operate menu then using the left mouse button for a Quick Pick option didn't work while using the right mouse to apply the Quick Pick option a group of curves did work.

Read

- **Case 25842**

If multiple models were selected to be read and one of them had already been read into T/HIS then the 'Extract curves to match model' option didn't work.

- **Case 25025**

In version 11 the option to search directories recursively for results only reported the 1st model it found in each directory of the directory contained multiple runs.

Version 12 now find and displays multiple models in the same directory.

- **Case 24652**

T/HIS incorrectly offered x,y and z moments as available data components for nodal velocity boundary conditions and not for rigid body velocity boundary conditions where they should have been available if data was read from the ASCII BNDOUT file.

If data was read from the BINOUT (LSDA) file then the correct components were displayed but if a FAST-TCF script was generated then it would incorrectly try and read moment data from nodes instead of rigid bodies.

Scripting

- **Case 23404**

T/HIS could get stuck in a loop if a JavaScript was run and T/HIS and a dialogue input command was used to try and read in a set of results that didn't actually exist.

```
DialogueInput('/MODEL READ', 'D:\\veh_21.key');
```

This has been fixed in version 12 but a workaround in version 11 is to use this instead:

```
DialogueInput('/MODEL READ', 'D:\\veh_21.key', '/');
```

T/HIS link

- **Case 25929**

If a D3PLOT model was closed while using the D3PLOT->T/HIS link and then reopened the 'Pick Visible' option for picking entities interactively in the T/HIS read menu would not be available.

- **Case 25070**

When running the D3PLOT->T/HIS link T/HIS could crash if a model was read into just T/HIS and the user selected the option to automatically extract curves to match some already read in from another model.

- **Case 25014**

If lots of curves were transferred from the XY-PLOT option in D3PLOT to T/HIS and the number of curves in T/HIS went past 1000 then the '+' and '-' buttons in the T/HIS curve manger that are used to map curves in chunks of 1000 at a time were not activated so you couldn't get to any curves past 1000.

(A work round in 11.0/11.1 was to create a single curve using any of the curve operations as this would then update the Curve Manager correctly).

- **Case 24384**

If when running the D3PLOT->T/HIS link the option to undock the T/HIS menus (Curve Manager, Operate, Maths, Seismic, Automotive) was used then the undocked menus would initially be the same width as the docked ones. This was different to the standalone version of T/HIS where undocked menus are initially twice the width of the docked ones.

In version 12 the undocked menus in the linked version are also now twice the width of the docked ones when they are initially undocked.

- **Case 23486**

If the directory selector was used inside T/HIS to select a directory to search for curve files in when using the D3PLOT->T/HIS link then both T/HIS and D3PLOT would lockup and stop responding to the user.

User Interface

- **Case 25845**

The default file filter in T/HIS has been changed from '*.*' to '*' when browsing for results files.

On a PC '*.*' showed everything including files without extensions so you could easily select 'binout' if you wanted to. On LINUX the file selector would only show files with extensions if the filter was '*.*' and this made it difficult to select 'binout'.

'*' works fine on both and shows all the files.

2.3.2 Bugs Fixed in 12.1

Datum

- **Case 26968**

A new Fast-TCF script command, 'datum remove', has been added to remove datums from graphs.

Fasttcf

- **Case 27631**

T/HIS 12 could crash if the curve history option was used to update a curve and the list of curves in T/HIS had not been condensed to remove any gaps in the curve numbering.

- **Case 27185**

When running a FAST-TCF script the commands to regenerate a curve using the 'keyboard' option could fail as T/HIS would prompt the user to overwrite an existing curve.

Graphics

- **Case 27853**

Autoscale did not always fit all the curves on the graph if some curves used the Y2 axis and Y2=0 was aligned with Y=0 on the first axis.

- **Case 26462**

If the option to turn on the 2nd y axis was used when no curves were selected for the 2nd y axis T/HIS could hang if either the minimum or maximum axis values for the 1st y axis was 0.0

Groups

- **Case 26813**

In version 12 it was not possible to read data for Part Groups for any model apart from the 1st model read into T/HIS. The entity menu would correctly show the Part Groups in all models and allow you to select them but the data for any model apart from model 1 wasn't read in.

Measure

- **Case 27942**

A change to the border width in the measure menu wasn't updating the graphics immediately like all the other options. It now does.

- **Case 27840**

Measurements were not being drawn in the correct location if one or both of the curves used the 2nd Y-axis. Using a logarithmic scale for the axis (either of the 1st or the 2nd Y-axis) would also cause the measurement to be drawn in the wrong position. The values reported were still correct.

- **Case 26680**

It was not obvious how to resume measurements if the user pressed 'q' for quick pick mode.

When 'q' is pressed, the measure panel will now only show the top 'Measure' and 'Annotate' buttons, and will change to either 'Resume Measure' or 'Resume Annotate'. These should be pressed once the user has finished their quick pick operations.

If the legend was floating and overlayed a curve, the cursor changed to make it look like you could resize/move it, but clicking still caused a measure pick. The cursor now stays on the '+' icon.

- **Case 26678**

A vertical scroll bar was missing if there wasn't enough space to show all of the measure panel. It has now been added.

Menus

- **Case 27946**

If the model database option was used in T/HIS or D3PLOT then some actions would results in spurious but harmless 'DID NOT PROCESS COMMAND' messages being generated in the window the application was started from.

- **Case 27941**

If when reading in data for solids elements the user chose the option to select data at different integration points then on returning to the 'Select Solids' menu the title of the menu listing all the available solids was incorrectly set to 'Select Surface(s)'. This didn't affect anything but was wrong.

- **Case 27630**

If you were using multiple pages in T/HIS and you had set a background image for some of the graphs then the 'Background Image' menu didn't update to show the settings for the graphs on the current page when you changed page.

- **Case 26939**

If the command line option to create a new graph was used

/layout graph create

or the FAST-TCF command

layout graph create

after the user had changed the value for the 'number of graphs to create' in the Graphs menu then T/HIS would create more than one additional graph. This meant that FAST-TCF scripts could create different numbers of graphs depending on the settings in T/HIS when the scripts were run.

The command line and FAST-TCF command have been modified so that they always create 1 new graph.

Read

- **Case 27924**

If a curve was marked as a NULL curve and was then subsequently deleted the internal 'null' marker wasn't cleared. If a new curve was then created with the same ID it wouldn't be plotted as T/HIS would still think it was a NULL curve even if it did now contain valid data.

- **Case 27910**

If you had a model which had fully integrated shells for which data at multiple in-plane integration points had been output to the THF file T/HIS could incorrectly think that some of the in-plane integration point data was invalid and would generate NULL curves. This problem would occur if the shell element had failed at only some of the in-plane integration points.

- **Case 27826**

If a model containing nodal rigid bodies was run which didn't use arbitrary material numbers then T/HIS could not calculate the position of part energies and velocities in the THF file correctly. T/HIS 12.1 has been modified to try and work out this position using additional information from either the ZTF or LSDA (binout) files.

If T/HIS 12.1 fails to calculate the position the PART data is now marked as unavailable in the THF file. This has been done to ensure that data isn't read and incorrectly displayed.

- **Case 27749**

T/HIS could crash if the user attempted to read data from a THF file which had more than 1,000 data variables written out for element. Version 12.1 had been modified to allow up to 10,000.

- **Case 27612**

If you read in lots of curves into T/HIS 12.0 using the ISO curve option then it was very slow as T/HIS would redraw the graph and update the curve menu after each curve was read.

In version 12.1 a message is displayed in the dialogue box as each curve is read and the graph and curve menu is just updated once after the last curve.

- **Case 27182**

T/HIS would fail to read an ASCII RCFORC file if any of the contact names contained the word 'slave' or 'master'.

- **Case 26833**

T/HIS would fail to display discrete beam elements written out to the DISBOUT file (*DATABASE_DISBOUT) if output hadn't also been turned on for the ELOUT (*DATABASE_ELOUT) file.

- **Case 26717**

If T/HIS was reading contact forces from an LSDA (binout) it was possible for T/HIS to incorrectly label the curves if the contacts had not been written to the LSDA file in order of increasing ID.

- **Case 26674**

In version 12.0 T/HIS could crash if 'drag and drop' was used to open a set of LS-DYNA results. The crash was random and didn't always appear.

- **Case 26599**

T/HIS 12.0 could fail to read contact forces from the ASCII version of the RCFORC file.

When the contact forces menu was displayed the menu wouldn't display any contacts.

- **Case 26385**

T/HIS could crash if the option to reread a model was used and data for rigid walls had been written to the LSDA (binout) file.

Settings file

- **Case 26634**

If a setting file was written out from T/HIS and either a graph title or an axis label contained spaces then the setting file wouldn't read back in correctly. The problem was caused by the title and axis labels not being enclosed in "s when the settings file was written out.

A workaround for version 12 is to manually edit the setting file and add the pairs of "s.

T/his link

- **Case 27007**

T/HIS would crash when running the D3PLOT->T/HIS link if the user right-clicked on a curve that had been read in from a model that was only loaded into T/HIS (opened using the 'Open New Model' option in T/HIS).

Units

- **Case 26996**

If a curve was read in from a data file with no units defined and then a curve from an LS-DYNA model was read in with units defined, then no units were displayed on the graph axes.

If a curve has come from a file with no units defined, the units of the model are now used.

2.4 REPORTER

2.4.1 Bugs Fixed in 12.0

Capture

- **Case 24933**

In the D3PLOT object command file, extra commands longer than 80 characters did not work. This has been fixed.

Data access

- **Case 25630**

The Job File option for T/HIS objects can now be left blank. This is useful if you want to read results from curve files, csv file or the keyboard rather than an LS-DYNA analysis. Previously REPORTER would not allow this.

Fasttcf

- **Case 24979**

FAST-TCF objects did not use the job filename when generating. FAST-TCF would look in the directory for the job. If there were multiple jobs in the same directory it could choose the wrong file. This has been fixed.

Graphics

- **Case 25848**

Circles and ellipses were not rendered correctly in pdf files. Now fixed.

Image

- **Case 24988**

REPORTER could crash if a library image was inserted very near the top or right hand edge of a page. This has been fixed.

Pedestrian

- **Case 24491**

The library scripts for contouring the bonnet with HIC values, 'pedestrian_zone_area_from_csv.js' and 'pedestrian_zone_area_from_variables.js', could be slow if they had to extrapolate a lot of values, e.g. when there were no impact points on the bonnet perimeter. This is now much quicker.

2.4.2 Bugs fixed in 12.1

Deleting

- **Case 26416**

If the oasys*temp_dir preference was set REPORTER would delete files from that directory when it exited. Now fixed.

Output

- **Case 27640**

When combining reports if one of the reports to combine did not exist hyperlinks could link to the wrong pages.

This has been fixed.

Pages

- **Case 27629**

The GTR pedestrian library pages would stop with an '%%% ERROR %%% No Result files found' message. This has been fixed.

Pedestrian

- **Case 27607**

The GTR pedestrian head impact template did not work on UNIX due to image filenames being a mixture of upper and lower case. This has been fixed.

Powerpoint

- **Case 26906**

Ampersand characters (&) in table objects created corrupt pptx files that could not be opened. This has been fixed.

Standard templates

- **Case 27430**

The EuroNCAP and GTR head impact standard templates did not work correctly if the node for extracting data was specified using the DATABASE_HISTORY_NODE_ID name, rather than the ID number. This has been corrected and will now work with either.

- **Case 26911**

The standard templates were calculating the Tibia Index incorrectly. The resultant moment was being calculated using the My moment twice. It now uses the Mx and My moment.

This affected the following templates:

CNCAP ODB 2012
EuroNCAP ODB (2014 and 2015)

IIHS ODB
IIHS SOB

- **Case 26724**

When processing neck extension, the standard templates should be looking at the peak -ve bending moment, but they were looking at the peak +ve bending moment. This has been fixed.

Variables

- Case 27256

When writing a reporter_variables file after generating a report variables that were expressions were not evaluated. Now fixed.

2.5 Shell

2.5.1 Bugs fixed in 12.0

General

- **Case 24278**

If a list file was used to submit a number of jobs then the Shell was ignoring the memory to use for each job if it was specified in the keyword file with '*KEYWORD xxxxxx'. Now fixed.

User Interface

- **Case 24277**

The Dyna submission window and program option windows did not have an [X] button in the top right hand corner to close them.

2.5.2 Bugs fixed in 12.1

General

- **Case 26970**

In version 12.0 on LINUX the PRIMER process could get stuck in a loop printing lots of 'OK' messages if there were no licenses available and it had been started from the Oasys Shell with the option not to use an 'xterm' set.

3 Enhancements

The enhancements and bug-fixes for each program have been broken down into a series of topics. Within each topic enhancements and bug-fixes are listed by case ID (most recent first).

3.1 PRIMER

3.1.1 Enhancements in Version 12.0

Adhesive

- **Case 23804**

A new adhesive type has been added allowing the user to create 'patches' of adhesive based on a source mesh.

Airbag

- **Case 13579**

When a *AIRBAG_HYBRID(_JETTING(_CM)) definition is converted to type *AIRBAG_PARTICLE PRIMER now attempts an 'intelligent' conversion between the two definitions.

In summary set ids, loadcurves, gas information and jetting (if present) are transferred from old to new definitions. The transfer is a long way from complete because the two airbag types are very different in detail, but it preserves a reasonable amount of data.

A warning message is printed when the conversion is performed explaining what was done in more detail.

Animation

- **Case 22072**

Animation of BOUNDARY_PRESCRIBED_MOTION definitions should now be faster, particularly for relatively simple definitions and for fine meshes.

- **Case 21617**

The animation function for BOUNDARY_PRESCRIBED_MOTION and BOUNDARY_PRESCRIBED_FINAL_GEOMETRY now includes repeat/play once options

Belts

- **Case 24899**

The following changes and enhancements have been made to the seatbelt fitter:

+ The distinction between a 'Pelvis' (free) slipring and a 'B-Post' (limited rotation axis) one has been made explicit.

Existing decks which just have a slipring defined in the belt path will work as before, but passage through the belt fitter will update them to mark the slipring as the appropriate type.

Making a slipring explicitly 'free' gets rid of the problems that users have encountered previously when trying to make belts with geometry other than 'retractor and slipring on b-post', since it will not make any assumptions about geometry or belt twist.

+ The definition of an 'acute' angle in the belt path is now user-definable.

By default an angle tighter (less than) 90 degrees is still considered to be acute, and will imply a new belt segment. However this angle can now be changed so, for example, making it smaller will allow belts to have tight loops of continuous mesh.

+ Transverse friction of the belt is now user-definable.

Previously belts had a built-in and fixed transverse friction coefficient (of approximately 0.2) that resisted sliding across a dummy once in contact.

This default is now zero, meaning that belts are more likely to slide and therefore to adopt the 'shortest distance between fixed points' path, but it can be defined as any value in the range 0.0 to 1.0 by the user.

+ Reduced transverse friction also means that the belt is more likely to 'fold back on itself', especially in the lap region. Therefore the default transverse limiting curvature angle has been changed from zero (unconstrained) to 30 degrees, effectively giving the belt path a little bit of transverse bending stiffness.

+ Default #iterations before assuming convergence has been increased.

In order to give a belt more time to slide across a dummy the default maximum number of iterations in both manual and auto-refit processes has been increased. If convergence occurs then fitting will cease as before, but if it does not the larger number of iterations that are now the default make it more likely that the 'shortest path' will be found.

+ Self-contact of the belt is now considered.

Historically the belt fitter calculated contact between belt and dummy, but not between overlapping sections of belt such as those close to the pelvis slipring.

When 1d belt elements were used close to sliprings this did not matter, but now that 2d belt elements are becoming commonplace this could give rise to penetrations if the belt was in a single-sided contact.

Overlapping sections of belt will now keep themselves apart, with the 'later' (higher numbered path points) section being further out from the dummy.

+ The titles of cross-sections added to a belt have been improved.

These have always had labels and titles giving their position on the belt path, but for cross-sections next to a retractor or slipring the title now gives the adjacent retractor or slipring number as well as the distance from it. This should make it easier to identify which cross sections should be used for which elements.

+ The overall length of the belt is now reported, and can also be assigned to a parameter.

Previously the length of each belt segment was reported as it was meshed, but the overall belt length (ie the sum of these) was not calculated. This sum is now reported as part of the post-meshing summary.

In addition a new ability to assign this length to a parameter has been added. This works as follows:

- The user designates a parameter (by name).

- This is a normal *PARAMETER definition, type real, that can be referred to anywhere in the model.

- When a belt is (re-)meshed the value of this parameter is updated to become the new total belt length, and any cards which refer to it are themselves also updated with the new value.

The intention is that this parameter should be used to define characteristics of belt-related items, for example seatbelt sensors, which depend on belt length and which can thus be updated automatically whenever the belt is remeshed.

+ The need to remember a segment set as part of the *BELT definition has been removed.

Internally PRIMER converts the elements and parts used to define the dummy into a segment set which is then used when calculated belt-to-dummy contact, and historically this segment set has been remembered as part of the belt definition.

There is no real need for this as the segment set is recalculated each time prior to fitting, and keeping it can present problems when a new dummy is swapped in because it may refer to nodes that no longer exist. Therefore this segment set, and the reference to it on the *BELT card, is no longer written out to a keyword file unless it is used explicitly in some other context - for example to define a *CONTACT.

+ Replace dummy has been updated and improved.

This feature was in PRIMER 11 but was difficult to use and thus of limited usefulness. It has been improved both functionally and ergonomically to the point where it should now be a useful aid to replacing dummies.

- **Case 24841**

In the seatbelt fitter the titles assigned automatically to database cross-sections now include the label of the adjacent retractor or slipring if they are at such points. This makes it easier to identify which cross-section belongs where during post-processing of data.

- **Case 24656**

When meshing a seatbelt PRIMER always reported the centreline length of each segment (torso, pelvis, etc) but not the overall length.

The total length, ie the sum of all segment lengths, is now reported as well.

- **Case 23000**

Some clients have complained that the belt fitter in PRIMER doesn't always give the 'correct' (whatever that is!) straight line fit across a dummy. What they mean is that it doesn't necessarily give the absolute shortest path; also that the final path can be very dependent on the initial points used to define the base path.

This is, at least in part, because the belt fitter has a pseudo friction model which limits movement at points where the belt is in contact with the dummy, emulating the effects of friction. In V11.x the effective coefficient is about 0.3, and this is hard-wired in with no means of changing it.

In V12 the following changes have been made:

- The friction coefficient now defaults to 0.0 giving a more 'slippery' fit that is more likely to give the shortest path, and will also be more likely to give a final shape that is not so dependent on the initial path point locations.

- The friction coefficient is now user-definable in the range 0.0 (slippery) to 1.0 (sticky). This can be changed interactively using fitting 'parameters', and is stored in the *BELT keyword definition.

Users wishing to revert to pre V12 behaviour should find that a friction coefficient of about 0.3 gives very similar results.

- **Case 22171**

A criticism of PRIMER's belt fitter is that it is hard to use when fitting a belt to a child seat. Several changes in V12 help with this:

- Explicit selection of slipring type avoids the unwanted twisting that could occur previously if PRIMER tried to fit a 'B post' slipring to an inappropriate location.

- Explicit control of belt orientation at slipring locations, especially by setting a vector from 2 nodes, makes it a lot easier to slot the belt accurately through gaps.

A worked example of fitting a belt to a child seat has been added to the user manual.

- **Case 22091**

The standard 'check' functions in PRIMER missed the following error:

Model with 2d seatbelt requires a node set EDGSET to be defined on the SECTION_SHELL card for the 2d belt elements. This set contained a node which was not on a 2d belt element which caused a failure to initialise in LS-DYNA.

A check for this has been added.

- **Case 21714**

There is a request to 'remove the restrictions on rotation of a B-Post slipring'.

Before V12 PRIMER automatically chose the slipring type (B-Post or Pelvis) from the adjacent belt geometry. From V12 onwards the user chooses between B-Post and 'free' sliprings, and the latter will give the freedom requested here.

Checking

- **Case 25452**

It is now possible to control whether or not master categories (eg Airbag, Dummy, etc) get checked during Model check via preferences in the oa_pref file.

This can speed up checking in large models by removing checks for things either known to be correct or which which will not affect an analysis.

- **Case 25311**

The speed of Model Check has been improved, and some of these improvements will also provide modest speed-ups in other PRIMER functions. A summary of the changes made is:

Internal 'flagging' functions, used heavily in model check but also generally, have been parallelised. This starts to show benefits for models with more than 1m elements, and becomes more significant as model size increases. At 20m elements it probably gives around a 20% speed up.

Contact checking, a heavy user of cpu time during model check, has had its existing parallelism further improved. Again, the benefits show up most clearly in larger models.

Some specific bottlenecks, for example nodal initial velocity and solid adhesive checking, have been recoded to run much faster with large models.

In a typical large model the improvement in speed of Model Check should be at least 20%, and where previously it was slowed down by particular bottlenecks that have been fixed the improvement will be better still.

- **Case 25147**

Joint coincidence checking (for N1N2, N3N4, etc pairs) has always been a bit of a problem because of a dimensionless tolerance of 1.0e-3 hard-wired into LS-DYNA.

From LS971 R7.1 onwards this 1.0e-3 check is now a warning rather than an error in LS-DYNA, and a new check (an error) of 1.0e-4 times the distance N1N3 has been added for non-spherical joints.

PRIMER handles this as follows:

If the output version is earlier than 971 R7.1 the 'old' logic applies, to wit:

- 1.0e-3 is a fixed dimension error

- 1.0e-6 x the model diagonal is a warning.

If the output version is 071 R7.1 onwards:

- 1.0e4 x the N1N3 distance is an error

- 1.0e-3 is a fixed dimension warning

- 1.0e-6 x the model diagonal is a warning.

In the post 971 R7.1 case the tests are applied in the order given here and, to cut down on unnecessary messages, if an error is detected then any subsequent warnings are not written. This is because failure of one of the criteria above is likely to fail the others too, and it doesn't really add to understanding of the problem to report it in multiple different ways!

- **Case 24290**

A new error check has been added for when and are both zero on the *ELEMENT_MASS_PART card.

- **Case 22175**

A warning check has been added if a node is referenced by a null (*MAT_NULL) shell/beam, and is not referenced by any anything else.

- **Case 22169**

A new optional warning check has been added to issue a warning if release flags are set on *NODAL_RIGID_BODY's. This is off by default.

Connections

- **Case 23802**

Support for reading VIP connection files has been added into PRIMER. Also, if the panels being joined is referenced by CAD name rather than CAE PID, PRIMER will look at the Part CAD name (from an imported BOM) or the Part heading to determine what to connect.

- **Case 23235**

Wildcards are now allowed when defining connection layers (Part ID type only) on the connections table. A '?' represents one digit, and a '*' represents any number of digits. So, a layer part ID of 10?? will reference any shell part in the model with a label between 1000 and 1099. A layer part ID of 10* will reference any shell part in the model with a label that begins with '10'.

- **Case 23027**

Up/down arrows will now scroll the selection on the connection table and if the mode is set to sketch or show weld and panels, that action will be automatically applied.

- **Case 22937**

In DEFINE_HEX_SPOTWELD_ASSEMBLY panel, added an object menu to select multiple solids at the same time and listed in ascending order in the list.

- **Case 22164**

A preference has been added for the user to specify valid spotweld material types (default is only MAT_SPOTWELD).

- **Case 21366**

Connections now store the name of the xml file from which they have been created. This information is displayed on the connection table under the header Xml Filename.

- **Case 21174**

A new option has been added to create connections using assemblies or part sets to define the parts joined together. When using this option, PRIMER will just connect to any shells within the specified group of parts in the vicinity of the connection point/line/patch.

- **Case 19629**

Master Connection Files (.mcf) can now be read and written by PRIMER.

Constraints

- **Case 22173**

New fast part pick and autocreate options have been added to the *CONSTRAINED_RIGID_BODY edit panel.

Contour

- **Case 24661**

Added 'Shell Normal/Align' button on Contour->Shell Normal->Setting panel. This will give direct access to Normal/Align menu

- **Case 24655**

The ability to colour the structure by material number (eg 20 for MAT_RIGID) has been added.

This is done by contouring: select SI/CT contour category Matl Props, component Material Number

Cut section

- **Case 20655**

The ability has been added to reverse the direction of a *DATABASE_CROSS_SECTION definition, or a number of definitions at the same time.

- **Case 15950**

PRIMER now considers *ELEMENT_SHELL offsets in the following contexts:

Cut-section graphical display, and also section properties display and calculation.

Normal graphics if the Display Options, shell true section switch is turned on to either flat or smooth. (This is not the default)

Contact penetration checks.

All of the above will consider offsets defined by NLOC on *SECTION_SHELL or *INTEGRATION_SHELL cards, also the _OFFSET suffix on the *ELEMENT_SHELL card.

Note that LS-DYNA, and hence also PRIMER, only consider offsets during the contact penetration calculation if:

Field CNTCO is set on *CONTROL_SHELL

The contact surface is not an AUTOMATIC type

In addition nodes on the slave sides of NODES_TO_SURFACE contacts do not 'inherit' the offset of their parent shell.

Notes:

1: This is nothing to do with the optional OFFSET suffix available for certain contact surface types.

2: For more information on how and where offsets are applied to contact surfaces see the LS-DYNA user manual remarks on data field CNTCO on card *CONTROL_SHELL.

3: Offsets during contact are not displayed by default in contact penetration plots, but if the 'as thick' option to show the thickness of contact segments is selected these will also show any offsets.

Database

- **Case 21778**

There have been several requests for PRIMER to return more memory to the operating system when it is no longer needed, thus reducing the process size. In particular people can be surprised that deleting a model does not always result in a corresponding reduction in the PRIMER process size as reported by the operating system (ie by Task Manager on Windows, or 'top' on Linux).

PRIMER does in fact have strategies for returning memory, but in order to be robust it will only do so if:

+ There are no models in memory.

and

+ All storage for a given data category (eg *NODE) has been returned to the memory allocator as 'no longer in use'.

The reason for this conservative approach is that if a piece of memory is accidentally referred to after it has been returned to the operating system then PRIMER will crash, and we prefer that not to happen!

If only a subset of models are deleted, leaving one or more behind, then PRIMER will not return that memory to the operating system, but rather will retain it for future use. This retention is efficient: memory released from (say) nodes in model A will be reused for nodes when model B is read in; therefore repeated read/delete cycles for models will not result in excessive build-up of memory use.

You *can* persuade PRIMER to reduce its memory usage, but to do this you will need to delete *all* models in the database. However even then memory will not reduce to the level of a new PRIMER session, so if memory usage on your machine is tight it would be better to close the PRIMER session altogether.

Deleting

- **Case 22184**

For Remove->cleanup, the user can now specify the cleanup and individual include, or PRIMER assembly, rather than the whole model.

- **Case 18630**

Cleanup Unused has historically only applied to a whole model, but this has been extended so that it can also be applied to:

Selected include files.

Selected part tree assemblies.

If either of the above are chosen then only items in the selected include files or assemblies will be eligible for cleanup. (The ability to clean up a whole model still remains.)

This capability is available from the Remove, Cleanup menu as before; also from the Part Tree (include files and assemblies), and from the Include tree (include files only).

Dummies

- **Case 25492**

A new option has been added to Dyna dummy positioning to remove rigid locks between assemblies. Sometimes rigid body constraints are created in the model across assemblies to simulate the locking of assemblies relative to each other during a crash test. During positioning these would not be locked so should be removed during the setup of the pre-simulation analysis model.

- **Case 25366**

Added a new option in dyna dummy positioning. When on, PRIMER will look for CONSTRAINED_RIGID_BODY definitions that span assemblies in the dummy and deletes them. This is to account for these rigid connections found some dummies that you would want in the full analysis, but not during positioning.

- **Case 23023**

'Soft' stop angles have been added to the dummy positioning tree file.

These are optional and provide a means of limiting articulation of joints to +ve and -ve angles that are less than the limiting 'hard' stop angles defined on the *CONSTRAINED_JOINT_STIFFNESS card. This can be useful when utilising the full 'hard' angle range results in initial penetrations since a smaller 'soft' angle can be set to limit articulation so as to avoid these penetrations.

'Soft' angles are set in PRIMER's dummy tree file, which is placed after *END, so the subsequent LS-DYNA analysis does not 'see' these values which only have significance during pre-processing.

Edit

- **Case 24270**

A new tool 'Text edit' has been added.

This is a generalised version of the function of the same name available from edit panels, but this will work on any keyword in PRIMER with a couple of exceptions listed below.

The keywords, which must already exist, are exported to a mini keyword file and an editor is opened on that file. The user can then edit this file manually, and can also add new entries to it.

When editing is complete the user closes the file and PRIMER reads it back in. Existing definitions read in this way supersede the originals, equivalent to normal PRIMER editing. New definitions are added to the model as if they had been read in normally from a keyword file.

The keywords that currently cannot be processed in this way are:

*INCLUDE and *INCLUDE_TRANSFORM

*CASE

The reason is that these are extremely complex to process, especially in conjunction, and the simple 'read and update' process is not sufficient to handle them.

- **Case 23795**

Historically the 'Copy Existing' button in an editing panel has only offered for selection other items of this type in the same model as the item being edited.

This restriction has now been relaxed, and items of this type in any model in the database can be copied.

Elements

- **Case 23349**

Added a 'Reverse' button in the Element->Beam menu to swap nodes N1 and N2 of the beam elements

- **Case 17293**

Added edit panel and checks for *ELEMENT_DIRECT_MATRIX_INPUT.

- **Case 14005**

Added the create/edit panel for the *Element_Direct_Matrix_Input card into PRIMER

Graphics

- **Case 25097**

SPH elements in PRIMER have historically been drawn as spheres of a fixed size (20 screen units).

Now if enough information is available to compute their true radius this is used instead. It requires the mass/volume field on the SPH element card to be non-zero, and if a mass is supplied then the material density must also be known.

If these data are not available the graphics reverts to the 'old' logic of a fixed radius sphere.

- **Case 23816**

The ability to plot 'true' beam cross-section shapes was added to PRIMER 11, but this did not take into account any offsets. This has now been added, and offsets defined via:

*ELEMENT_BEAM_OFFSET (W vectors)

*SECTION_BEAM (NSLOC and NTLOC)

*INTEGRATION_BEAM (SREF and TREF)

are now all considered when beam true section display is turned on.

This also means that when a cut-section is made through beams and the 'properties' calculation is invoked the beam sections will include any offsets.

- **Case 22969**

An option has been added to turn off the display of the numbers on the graticule.

- **Case 22186**

Options have been addded to the colour menu to set the colour for *CONSTRAINED types - NODAL_RIGID_BODY, RIGID_BODY and EXTRA_NODES. This will aid the user in telling these items apart, for example during a find attached operation.

- **Case 15722**

Added sketch function to following keywords:

*DEFINE_STAGED_CONSTRUCTION_PART

*DAMPING_FREQ_RANGE

Few minor changes in sketch related functions to enable 'SKETCH' for following keywords:

*BOU_PORE_FLUID (draws the part or set of parts)

*BOU_PWP_TABLE (draws the part or set of parts)

*BOU_PWP_NODE or SET - (draws the node or node set)

*INITIAL_STRESS_DEPTH (or SET)

*INITIAL_PWP_DEPTH (or SET)

*LOAD_GRAVITY_PART (or SET)

*LOAD_STIFFEN_PART (or SET)

- **Case 12774**

There is now a Display Option to draw shells using their true thickness.

The default is still a single facet that is infinitely thin on the nodal plane of the shell, now referred to as 'thin'; but two 'thick' options have been added:

Thick flat draws each shell as a separate block, with no attempt being made to join up adjacent edges.

Thick smoot as above, but attempts to make a continuous surface over adjacent shells by joining up their edges.

The two 'thick' modes take into account the actual thickness of the shell, including any variation across nodes (T1 to T4). They also take into account any offset defines via *ELEMENT_SHELL_OFFSET, or by one of the NLOC arguments on *PART_COMPOSITE or *SECTION_SHELL.

If a shell is a composite, ie *PART_COMPOSITE or *INTEGRATION_SHELL with multiple layers, lines are drawn on the 'side face' showing each layer, making it possible to check thicknesses.

Finally if shell capping is used during cut-section display then in 'true' mode the cut section will show the actual cut shape through the true 3d shape of the shell element.

Include

- **Case 23740**

PRIMER 12 now supports *CASE in combination with *INCLUDE_TRANSFORM.

PRIMER 11 did not support this (we ran out of time during development) but this restriction is now lifted, and both *INCLUDE_TRANSFORM and *DEFINE_TRANSFORMATION cards may now be used inside *CASE statements.

Other transformation keywords *PART_MOVE and *NODE_TRANSFORM are also now supported inside *CASE.

- **Case 15912**

Added an option to read include stamped part as a new model in plastic strain view mode. It allows the user to read the stamped part by right clicking over stamped part in include part tree.

- **Case 12069**

Added include file selection/viewing feature on database edit panels.

Keyword

- **Case 24545**

New popup option '1.0' is added for _GEOMETRY cases.

- **Case 21378**

*RAIL_ keyword editing panel now supports 'Sketch' functionality. In *RAIL_TRACK, it sketches two beam sets BSETID1 and BSETID2 and in *RAIL_TRAIN node set NSETID is sketched.

- **Case 20649**

The popup in the IMFLAG field in the *CONTROL_IMPLICIT_GENERAL edit panel can now be used to select loadcurves

- **Case 19808**

A shortcut 'K' button has been added to edit panels that opens the LS-DYNA keyword manual at the relevant page.

- **Case 18976**

Added hover text to fields on *DEFINE_CONNECTION_PROPERTIES keyword

Kwd editor

- **Case 20669**

The keyword editor allows you to highlight multiple rows, and a change to a data field in any highlighted row is propagated to the same field in all other highlighted rows.

However if you choose [right click, edit] to edit one of the highlighted items this launches the editing panel and edits it, but when the edit is finished any changes are historically not propagated to other highlighted rows.

This lack of capability has been fixed. After the edit is completed any changes between the pre-edited and post-edited definitions are found, and are propagated to other highlighted rows. Such changes are any combination of:

- Changed data fields

- Changed keyword suffices (which may en/disable/change data fields)

- Changed include file membership.

Since these could lead to many changes in the model they are listed first, and the user is required to choose whether or not to apply them.

Macro

- **Case 23181**

A new function MacroFunction() has been added to JavaScript which will return the equivalent macro commands for events such as button presses, popups etc.

Materials

- **Case 25487**

When using the material editor to change the type of a material, for example from MAT_NULL to MAT_ELASTIC, PRIMER will attempt to copy over data from the old material to the new where the name of the data field is the same. For example density RO in the old material will be copied to field RO in the new one, even if it is in a different location on the card.

This process depends on the names of data fields being the same so it fails where different names have been used for the same data on different material types. A typical example is Young's modulus which is 'E' on some materials but 'YM' on others, and hence does not get copied.

Therefore a list of alias names has been added which permit data fields to be copied if:

- Both old and new materials match one of the alias name pair

- The units of the two data fields are the same.

At present this only matches 'E' and 'YM' with the proviso that both must use the same Modulus units. Further pairs will be added in the future if and when they become apparent.

- **Case 22336**

Added the ability to filter selection of *MAT and *PART in object menus by material option _ADD. Currently the options are:

*MAT_ADD_AIRBAG_POROSITY_LEAKAGE

*MAT_ADD_EROSION

*MAT_ADD_PERMEABILITY

*MAT_ADD_PORE_AIR

(There are other _ADD suffixes under *MAT, but they refer to parts and not to materials.)

- **Case 21299**

Added pref for Max strain for curves with ETAN option with default 1.0.

- **Case 20654**

Both the scalar and the keyword editors for materials now offer drop-down menus to select restraint codes for the rigid material fields CMO, CON1 and CON2. This saves having to remember or look up the meanings of the various options.

- **Case 9873**

Drop-down menus have been added for the CMO, CON1 and CON2 fields on the MAT_20 (rigid) editor panels.

Mechanism

- **Case 16722**

The ability to move *DATABASE_CROSS_SECTIONs with Mechanism or Dummy assemblies during positioning has been added. The details are:

- Prior to positioning a scan of all *DATABASE_CROSS_SECTIONS is performed. Only _PLANE types with a finite part set PSID are considered.

- A cross section is associated with an assembly, and is moved with it if one or more parts within part set PSID have been used to define that assembly.

- However if a cross section contains parts that are used in more than one assembly it is not possible to tell which it should be associated with, so the section will not be moved. In this situation the user is given the option of 'cloning' the section into as many definitions are required to make them unique to all parts that the original spans, and if selected these sections *will* be moved with their respective assemblies.

- Parts that are shells using *MAT_NULL are ignored for the purposes of determining usage in assemblies, since null shell parts are often used both to coat dummies, hence spanning several assemblies. These parts contribute little or no force, and it would be annoying if they were treated as errors.

- Movement of sections with assemblies is switchable at the Mechanism / Dummy level, and the status of this switch (default = on) is stored in the keyout file.

Menus

- **Case 23673**

Visualisation of *BOUNDARY_PRESCRIBED_FINAL_GEOMETRY definitions can now be toggled on or off from the PRIMER Entities menu.

Mesh

- **Case 23712**

All meshing functions consolidated under one Mesh tools panel.

- **Case 20981**

Added 'create true thickness solid from shell' option to shell mesh extrusion.

- **Case 17796**

When remeshing an area or creating a hole PRIMER could create facing triangles in the mesh. The algorithm has been changed and this should happen less often.

Model build

- **Case 17712**

PRIMER now has a preference that will enable the writing of all includes (rather than just the master file) to build directories during a model build operation using csv files

Nodal rb

- **Case 23493**

Added a 'COMBINE' button in the main NRB(Nodal Rigid Body) docked panel to combines multiple NRBs into one. Parent nrbs are deleted and a panel will pop to select model in case of multiple models to combine NRBs.

The new NRB and new SET_NODE are created in layer (include file) obtained by following rule:

1. If all selected NRBs are in the same include, create the new NRB in the same include

2. If not, create the new NRB in the current layer.

3. Using the same logic for new SET creation.

Node

- **Case 17811**

A starting or seed node label can now be specified for those creation methods that could spawn multiple nodes - line, surface, geometry points, geometry curve. Previously, these labels would be auto-created.

- **Case 14498**

Added post end keyword *TARGET_MARKER with large label support

Orient

- **Case 23175**

The orient translate and rotate function is now available on the command line.

Output

- **Case 26114**

There is a special OEM version of LS-DYNA 971 R6.1 which looks at the data field W-MODE on *CONTROL_SHELL, something that is not normally written in standard 971 R6.1 format.

PRIMER will read this data field unconditionally, but if standard 971 R6.1 output format is selected it will not write it out, since the original version of LS-DYNA did not specify the required extra lines on *CONTROL_SHELL.

The preference 'allow_wmode_r61' has been added, and if this is set to TRUE then the extra lines and data field W-MODE will be written on *CONTROL_SHELL in 971R6.1 format.

- **Case 25341**

The existing Model Write d3plot/PTF file capability only write the basic control block and element topology of the model.

This has now been extended to give the option of writing 'state 1' of data, which will still be based on the undeformed nodal coordinates but will include:

- Shell thicknesses

- Any *INITIAL nodal velocities

- Any *INITIAL nodal temperatures

- Any *INITIAL stresses or strains for solids, shells and tshells.

This will also provide somewhere to display user-defined data, should users wish to use this file as a 'chassis' for displaying that.

- **Case 24541**

It has become a problem that the output formats of *BELT, *DUMMY and *MECHANISM in V12 contain extra cards and are thus not readable in earlier PRIMER versions. So a preference

primer*mdumm_keyout_version: v11 | v12 | current

has been added.

The default is 'current', but setting this to 'v11' will generate output in the PRIMER 11 format. A warning will be issued if any of the data fields omitted are non-zero.

Parameter

- **Case 24657**

Previously the top level parameter panel, which shows a summary of all parameters in the model, only displayed 10 rows regardless of the panel height.

This has now been amended and as many rows as will fit into the panel, up to a maximum of 100, will now be shown. Therefore making the panel taller will display more parameters.

In addition a column to show the include file in which the parameter definition lives has been added. This is (deliberately) narrow so a long filename will be truncated, but hovering over an entry will give a popup showing the full filename.

Part tree

- **Case 8772**

A red/green indication has been added to the part tree showing blanking status of parts/includes.

Pedestrian

- **Case 24931**

Hard point detection has been added to the pedestrian markup script.

The script can automatically detect and create impact points close to user selected hard parts. Alternatively points can be created manually by picking on the vehicle surface or on the hard parts.

A contour plot can also be drawn on the vehicle outer surface to indicate the proximity to the hard parts.

- **Case 24648**

An option has been added to the pedestrian markup script to create a circle of points around the automatically and manually created points. This can be used to do sensitivity studies.

- **Case 24641**

The pedestrian markup script can automatically calculate the angle, velocity and mass values for the upper leg impactor, but it assumed the model units were mm,kg,ms. An option has been added to use mm,T,s instead.

- **Case 23988**

For the GTR protocol the pedestrian markup script will now put impact points on the impact zone boundary lines when it generates a grid of points.

- **Case 23982**

A log file now gets written out by the pedestrian markup script when the csv file containing the impact points is written out.

The log file contains information on:

- The inputs used to create the lines

- The X,Y,Z coordinates of the impact points

- The velocity, target mass, bonnet height and bumper lead for upper leg impact points.

- **Case 23706**

The EuroNCAP v7.0 protocol (2014) has been added to the pedestrian markup script.

- **Case 22248**

An option has been added into the pedestrian markup script to label the impact points using the 'GM style'. This uses an 8 digit label that encodes information about the impact point. The default is still to use the method defined in the protocol.

- **Case 21927**

An option has been added to the pedestrian markup script to find head impact points that could potentially impact on to hard parts below the surface of the bonnet.

- **Case 21456**

The protocols state that if the bumper is identifiable the upper bumper reference line should only be marked on those parts. Previous versions of the pedestrian markup script did not allow for this option and the reference line would be marked on the parts selected for all the other reference lines (i.e. it assumed that the bumper was not identifiable).

A new button has been added to select bumper parts. This is optional and if nothing is selected then the line will be marked on the parts used for all the other reference lines (default).

- **Case 19154**

PRIMER's model-build-from-csv function now includes two new master style options:

- 'GM style' generates one user-defined file to which all *DEFINE_TRANSFORMATION definitions are written. Each *DEFINE_TRANSFORMATION definition is given a unique label

which is equivalent to the directory name, if valid. Each master model then refers to the appropriate *DEFINE_TRANSFORMATION by its label. Also, a couple of extra cancelling transformations are added such that the last translation represents the aim point. Additionally, a title string may be specified. The tile for each *DEFINE_TRANSFORMATION card would be a combination of the label, the aforementioned string, and the depenetration type.

A new 'First impact point only' option has been added which is specific to the GM style. This option will generate *DEFINE_TRANSFORMATION information (single file as above) for all loadcases but will only write out the first master model.

- 'CASE style' generates a single user-defined file to which all data is written. No additional master models are generated. Data for each loadcase - including *DEFINE_TRANSFORMATION - is specified under a distinct *CASE definition. *INCLUDE for the model and *INCLUDE_TRANSFORM for the impactor are only written once for the process.

- **Case 17126**

A new model-build-from-csv option has been added for the PEDHEAD build type that will permit target point projection along the line of flight. The old option of projecting vertically along Z is still available.

Also, 'Sketch' and 'Pick' in the interactive panel will now show the original point and the projected point.

Preferences

- **Case 17753**

The names of the preference files that are read are now written to the dialogue box when editing preferences

Read

- **Case 25493**

Recent files that have been read/written within PRIMER are now available via a dropdown on the file read input text box.

- **Case 17929**

In version 12 of PRIMER models can be opened on a PC by using drag and drop. PRIMER assumes that any files that are dragged and dropped into the main application window are LS-DYNA keyword files and will automatically try to open them.

Renumber

- **Case 25077**

It is now possible to renumber entities into any include ranges set for individual include files in the include renumber panel rather than having to renumber all include files.

- **Case 23114**

A new option has been added so that under normal operation PRIMER will avoid label clashes between entities which may lead to problems reading models into other pre-processers. For example, labels clashes between different element types.

- **Case 17771**

The include renumbering panel now provides additional information when there is a clash between label ranges of various include files

Rigid bodies

- **Case 24109**

An option has been added to the *CONSTRAINED_RIGID_BODY edit panel to swap the slave and master parts (PIDS and PIDM).

Scripting

- **Case 24682**

The position of the picking window can now be controlled from JavaScript by using Options.pick_window_position

- **Case 24651**

A File.Rename method has been added to enable files to be renamed.

- **Case 23747**

A new -js_arg command line argument has been added to PRIMER to enable arguments to be passed to a JavaScript.

- **Case 23508**

Model.Write() and Include.Write() now have an optional 'large' argument to allow large formal files to be written. There is also a new Model method UsesLargeLabels() that will return true if a model uses large labels.

- **Case 23504**

WidgetItems now have an index property which is the index of the WidgetItem in the 'parent' Widget (or undefined if not assigned to a Widget)

- **Case 23500**

Blank and Unblank methods have been added for the keyword classes (e.g. Shell and Part)

- **Case 23498**

An onBeforeShow event has been added to the Window class.

- **Case 23497**

A new static Window method MasterResolution() has been added to return the resolution of the master programme window.

- **Case 23420**

A new Tshell class has been added to support thick shells

- **Case 23179**

A new Belt class has been added to JavaScript to allow seatbelt definitions to be created and edited from JavaScript.

- **Case 23177**

Connections can now be created, modified and deleted for Mechanisms in JavaScript.

- **Case 23176**

A new WidgetItem onMouseOver event has been added.

- **Case 23078**

A new static method Widget.StringLength(string) has been added that will return the length of a string in Widget units.

- **Case 22775**

A (read only) window property has been added to the Widget class so it is possible to find which Window a Widget is defined in.

- **Case 22355**

It is now possible to get/set point data and get assembly info in the Dummy and Mechanism classes.

- **Case 22347**

An onAfterShow event has been added to the Window class.

- **Case 22345**

A new showClose property has been added to the Window class to control whether the close (X) button is shown on the top right of the window.

- **Case 22344**

A new Window.Delete() method has been added which frees any User Objects in windows. This should not normally required and is only required for scripts that (re)create lots of windows.

- **Case 22343**

A new Widget.Delete() method has been added.

- **Case 22342**

An active property has been added to the Window class to (un)restrict input in a window.

- **Case 22341**

Property keepOnTop has been added to the Window class to allow users to keep a window on the top of the stacking order.

- **Case 22225**

A listbox widget has been added to JavaScript.

- **Case 21998**

Classes have been added to JavaScript for pretensioners, retractrors, seatbelts, sensors and sliprings

- **Case 21642**

ItemAt() and TotalItems() methods have been added to the Widget class to give access to individual WidgetItems defined in the Widget.

- **Case 21596**

File.FindFiles() can now find directories as well as files.

- **Case 20274**

Added java script support for keyword *DEFINE_CONNECTION_PROPERTIES

- **Case 19830**

The list of available scripts is now shown as a tree and the scripts can be organised into branches

- **Case 18984**

Support for *DAMPING_GLOBAL has been added to JavaScript

- **Case 18905**

Added 'Ejection mitigation' Script, calculating target locations on window (Forward/Rear) based on FMVSS 226 specification and export as csv file, there is an option to build model from csv file.

- **Case 18756**

Include file comments can now be edited from JavaScript.

- **Case 18748**

Methods TableProperties, AllTableProperties and FlaggedTableProperties have been added to the Part class to allow the user to get all of the part table properties for parts.

- **Case 14189**

The list of available scripts is now shown as a tree and the scripts can be organised into branches

- **Case 13592**

Added an element SPH class into PRIMER JavaScript.

Seatsquash

- **Case 24539**

Added -2 and -1 as valid ELFORM types when creating *INITIAL_FOAM_REFERENCE_GEOMERTY through seatsquash.

Sections

- **Case 23587**

Added *SECTION_SHELL type 52 and 54 to the drop down list on the section shell edit panel.

Selection

- **Case 24986**

When selecting elements by screen area PRIMER has historically used the test 'is the element centre in the area' to determine whether or not the element is selected.

This remains the default, but an option to change this test to 'element will be selected if any node is in the screen area' has been added.

- **Case 24316**

PRIMER 11 could be distinctly sluggish about screen-picking when model size exceeded ~10 million nodes and elements, and testing with a model of ~50m elements revealed it to be unacceptably slow.

Picking in PRIMER 12 has been optimised to improve this, partly by parallelising more of the screen-picking process, but also by improving the underlying algorithms. The difference won't be noticeable for small models, but for these very large models it is now 10x or more faster and much more usable.

- **Case 22803**

Materials can now be filtered in object (selection) menus by the various _ADD options (erosion, etc)

3.1.2 Enhancements in Version 12.1

Airbag

- **Case 27402**

_DECOMPOSITION flag for *AIRBAG_PARTICLE is now supported.

Case

- **Case 27294**

Previously the order in which CASE_BEGIN/END pairs appeared could be a bit random. The file was correct, but hard to read because it was not in the expected order. Also reading a file, writing it out and then reading it in again could give different (but still valid) results.

This has now been made more consistent as Case begin/end pairs will always appear in ascending numerical order, and this order will no longer depend on the order in which they were read.

Checking

- **Case 26966**

Model check now gives a warning if constrained contact (tied or spotweld) applies to nodes or segments which belong to a part used in a deformable to rigid switch as this may result in unwanted behaviour, e.g. spotwelds becoming unstuck.

Connections

- **Case 27583**

A preference has been added to stop Primer automatically creating a set of MAT 100 solid elements it can't automatically form a connection from (due to topology reasons or no shells close enough to attach).

Output

- **Case 27035**

When PGP encrypted data is found in an input deck in a context that is not associated with any keyword it is treated as being 'free-standing', and because its contents are unknown it is backed up during read in a separate file.

On output the contents of this backed up file are copied verbatim to the output file immediately before *END, on the assumption that the free-standing content of the encrypted data is not sensitive to its location in the input deck.

This strategy fails when the encrypted data contains *PARAMETER definitions which are referred to elsewhere in the deck, since these are 'referred to before being defined', which is an error during LS-DYNA keyword input (although PRIMER will tolerate it).

So to give more flexibility to this process two new preferences have been added:

(1) primer*freestanding_pgp_output_location

This defines where in the deck the PGP data block is written and can be one of 'top_of_file', 'after_parameters', 'after_includes' or 'before_end'. For backwards compatibility the default remains 'before_end'.

(2) primer*freestanding_pgp_output_method

This can be either 'embedded', which means that the PGP data remains as a block within the parent file; or 'as_include', which means that it is written as a separate include file called 'fs_pgp_data.key'.

The 'as_include' option is restricted to the master file only, so that child include files do not proliferate; however if include files are being merged into a single master file during keyout then all free-standing PGP blocks in all files are copied to this single file.

Pedestrian

- **Case 26349**

The EuroNCAP v8 protocol has been added to the pedestrian markup script.

Renumber

- **Case 27568**

New options have been added to allow the user to switch off renumbering of materials and sections.

Seatsquash

- **Case 27656**

During importing a DYNAIN file, you now have the option to ignore the IALEGP field of *INITIAL_STRESS_SOLID cards. The value '1' is written to this field in some version of LS-DYNA (fixed in more recent versions) which can lead to errors being reported in PRIMER when they are not required.

3.2 D3PLOT

3.2.1 Enhancements in 12.0

Annotation

- **Case 14298**

Under Utitlities->Entity Names, names can now be defined for certain entity types and can be displayed on the model. This can be used to annotate the model.

These user defined names can be saved and reused in later D3PLOT sessions.

Cut Section

- **Case 25805**

In version 12 the cut-section menu has been redesigned to make it easier to access the different options available.

A new option has also been added that can be used to exclude entities if different types form the cut-section. If for example you have a block of reinforced concrete that is modelled with solid elements and beams it is now possible to exclude the beams form the cut-section so only the solids are removed.

Data Access

- **Case 23875**

D3PLOT now supports the output files from models containing *ELEMENT_DISCRETE_SPHERE's run in R7.1.

- **Case 22078**

The components 2D max shear stress and strain (S2MAX_2D_SHEAR_STRESS and E2MAX_2D_SHEAR_STRAIN) have been added.

Deform

- **Case 24017**

An option has been added to the DEFORM->REF NODE menu to output coordinates in the WRITE menu as [undeformed coordinate] + [displacement in local system].

Envelope

- **Case 21293**

The Contact menu did not have a button to create and Envelope plot. This has now been added in.

In previous versions, if you wanted to do an envelope plot of a contact component, you had to select the component under the Contact menu, then go back to the 2D/3D menu and select the Envelope button there.

General

- **Case 25309**

A new option has been added to write the parent directory and filename of the model to the graphics window header. This can be set in the top menu Display->Options.

Graphics

- **Case 25806**

In version 12 beam offsets are taken into account when beams are drawn using the 'True Section' option and a ZTF file written by PRIMER 12 containing the offset information is present.

- **Case 25361**

The number of decimal places shown for the time in the bottom right hand corner of the graphics window can now be controlled in Disp Opts->Window Dressing.

- **Case 23971**

The maximum number of characters displayed for the header can now be set in the 'Disp Opt->Window Dressing' menu.

Also, the following preferences have been added to control what is displayed in the graphics window:

```
d3plot*plot_header
d3plot*plot_contour_bar
d3plot*plot_clock
d3plot*plot_date
d3plot*plot_triad
d3plot*header_type
d3plot*header_nchars
```

- **Case 23878**

In version 12 the way that data values are displayed when using dynamic labelling has been changed. By default the values will be displayed using 'Automatic' formatting but if the user changes the format used to display the contour bar values then the same format will also now be used for the data values.

- **Case 22968**

An option has been added to turn off the display of the numbers on the graticule.

Measure

- **Case 19851**

Measurements can now be given a label which is displayed alongside the measurement in the graphics window.

Multiple Windows

- **Case 23761**

If models are open in multiple windows it is now possible to move them all into window 1. A new button has been added to the Window->Edit Window menu on the top menu to do this.

Nastran

- **Case 21865**

The list of loadcases written in the bottom left-hand corner of the graphics window from a Nastran analysis can now be turned on or off with the DISP OPT->NASTRAN LC LIST.

Preferences

- **Case 25631**

D3PLOT now has a '-pref' command line option that can be used to select a user defined preference file that is read after any other preference files.

```
-pref=filename
```

If 'filename' contains a PATH then it should be the complete path to the file, if it doesn't contain a PATH then it is assumed that the file is located in the installation directory.

- **Case 24983**

A preference has been added to set the cut section display type;

```
d3plot*cut_section_display.
```

It can be set to OFF (default), WIREFRAME or TRANSPARENT.

- **Case 24459**

A new preference

```
d3plot*open_models_in_w1
```

has been added to open models in Window 1 by default rather than the next window.

Read

- **Case 25505**

In version 12 of D3PLOT models can be opened on a PC by using drag and drop. D3PLOT assumes that any files that are dragged and dropped into the main application window are LS-DYNA output files and will automatically try to open them.

- **Case 25494**

Recent files that have been read within D3PLOT are now available via a dropdown on the file read input text box.

- **Case 24459**

A new preference

```
d3plot*open_models_in_w1
```

has been added to open models in Window 1 by default rather than the next window.

Scripting

- **Case 24362**

A GetCutCoords() javascript function has been added to get the coordinates where the cut-section cuts through an element.

- **Case 19649**

The GetNumberOf() function can now be called with the argument GROUP to get the number of groups in a model.

A new function GetGroupInfo() has been added to get information about a group returning an object with properties .label and .name.

These can be used in conjunction with the /GROUP dialogue commands to manipulate groups in a script.

- **Case 18492**

The javascript function GetElemsInPart() has been added to get the elements in a part.

User Interface

- **Case 23773**

In version 12 the maximum length for filenames (including the directory) has been increased from 256 to 512 characters

NOTE : Internally Windows supports filenames of up to 32768 but currently it does not support pathnames longer than 260 characters within Windows Explorer or the command line interface.

X Sections

- **Case 20832**

If X-sections have been named using the *DATABASE_CROSS_SECTION_ID option the names are now displayed alongside the cross section ID in the blank menu.

XY Plotting

- **Case 23803**

A new XY Data plotting button has been added to the quick pick menu when right-clicking (or dragging) on an entity.

An XY plot of the data component selected in the data menu is plotted for the selected entities.

ZTF File

- **Case 22116**

D3PLOT will now issue a once-per-session warning if MAXINT is not set to the default of 3 and the user is trying to plot by TOP, MIDDLE or BOTTOM surface and there is no ZTF file as in this situation D3PLOT may not be able to determine which is the correct integration point to read without the additional data in the ZTF file.

3.2.2 Enhancements in 12.1

Graphics

- **Case 27664**

Historically D3PLOT, and occasionally PRIMER, have suffered from jerky graphics performance when using the 'Out of the box' settings of the graphics driver on NVidia Quadro cards.

The exact reasons for this are kept secret by NVidia, but it is something to do with timing in the graphics driver itself and - in common with other engineering packages - NVidia have set up an 'Oasys LS-DYNA Suite' profile in their drivers which, if used, will fix the problem. However the graphics driver needs to “know” that it should use this profile instead of the default one.

Prior to V12.1 this has not been detected and set by default, meaning that the only solution is for the user to reset the default (global) NVidia profile on a Windows machine, meaning that this applies to all applications which do not auto-detect their own specific settings.

From V12.1 onwards this is no longer necessary: D3PLOT, and the other graphical Oasys Ltd codes, will detect that a Windows machine has an NVidia card installed, and will automatically detect and set the correct profile. No user intervention is required, and the “out of the box” settings for the NVidia driver can now be left unchanged.

3.3 T/HIS

3.3.1 Enhancements in 12.0

Annotate

- **Cases 17684, 20270 22156 and 23865**

Annotations can now be created on graphs (Measure->Annotate). They can be made on curve points or screen points.

Curves

- **Case 19108**

The curve 'History' option in Quick Pick now contains the option to change the model that data is being read from. Previously you could only change the entity ID and component before updating the curve.

FAST-TCF

- **Case 24591**

In previous versions T/HIS would output a command to label every curve generated by a FAST-TCF script and for most curves the command would usually contain the default curve label generated when T/HIS read in the data.

If when running a FAST-TCF the wrong data was somehow read in the user might fail to notice this as the default curve label that would show the incorrect data source would be overwritten with that written to the FAST-TCF script when it was generated.

From version 12 onwards T/HIS now keeps a track of the curve labels and when it writes a FAST-TCF script it checks to see if the curve label has been modified interactively by the user. If it has then the command to label the curve is written out as normal. If the label is still the default one generated by T/HIS then it is now output as a comment line only.

```
 #
 # Curve labels - (modified by user)
 #
 # label curve_12 K.E. - Part PART1002 : (knee Bolster springs)
 # label curve_13 K.E. - Part AAAAAAAA : (STEERING WHEEL RIM BEAMS @ STEERING
WHEEL RIM)
 # label curve_14 K.E. - Part 1100 : (arup bulk head beam)
 label curve_15 XXXXX K.E. - Part 1300 : (rigid top of steerin g column)
 # label curve_16 K.E. - Part 1302 : (STEERING WH RIM SHELLS @ STEERING WH
RIM SHELLS)
 label curve_17 XXXXX Disp x - Node 100437 : (Sled)
 # label curve_18 Disp x - Node 200000 : (Head)
 # label curve_19 Disp x - Node 100437 : (Sled)
 # label curve_20 Disp x - Node 200000 : (Head)
 # label curve_21 Disp x - Node 201786 : (Chest)
 # label curve_22 Disp x - Node 203303 : (Pelvis)
```

- **Case 22271**

T/HIS 12.1 has a new JavaScript function that can be used to get the value of a FAST-TCF variable.

```
var value = GetFtcfVar('job')
```

would return the value of the FAST-TCF variable 'job'.

(Note: all FAST-TCF variables are case insensitive).

- **Case 22259**

T/HIS 12.1 has a new JavaScript function that can be used to set the value of a FAST-TCF variable.

```
var value = SetFtcfVar('run_number','20')
```

would set the value of the FAST-TCF variable 'run_number' to '20'. If the variable already existed then it's value would be updated otherwise it would be created.

(Note: all FAST-TCF variables are case insensitive).

Graphics

- **Case 21065**

If a large number of curves were selected for a curve operation by dragging across them in a graph T/HIS could take some time to highlight all the curves in the graph and in the curve menu.

In version 12 this process has been speeded up.

- **Case 4246**

The Display menu now contains options to turn on and off the drawing of a graphs top and right hand axes. By default these are both set to ON.

As well as turning on and off the axes via the Display menu there now 2 new preference settings

```
this*axis_top: ON / OFF
this*axis_right: ON / OFF
```

Measure

- **Case 20672**

A new menu has been added to the tools panel for making measurements on graphs (shortcut key 'm').

- Multiple measurements can be displayed at the same time

- Measurements can be made between curve points and/or screen points

- Single points can be measured too

- The measured values can be displayed on the graph

- Labels can be added to the measurements

- There is an annotation mode for annotating your curves

- The formatting of the measurements are user configurable

Operations

- **Case 24032**

The Weighted Integrated Factor method has been added to calculate the correlation between two curves (menu Automotive->WIF).

- **Case 23254**

The Rolling Average function (R-AVE)has been modified so that it now has a user defined time window that it averages across.

By default the time window is set to 0.0 which means 'infinite' so the default output from the function is unchanged. If the user sets the window time to T, then for each curve point the y value is calculated by averaging between -T/2 and T/2.

- **Case 22899**

There are now two different scaling options that can be applied when doing an FFT.

Scaling option 1 - this is the default and recommended for most purposes. It is consistent with other signal processing software, giving a magnitude independent of any padding required to make the input curve have a number of points equal to a power of 2. Performing an inverse FFT on the resulting curves will NOT get back exactly to the original curve if it did not have a number of points equal to a power of 2.

Scaling option 2 - with this option, applying an inverse FFT to the resulting curves will generate a curve the same as the original even if it did not have a number of points equal to a power of 2. This is useful if users wish to create their own filters, where the filter characteristic is defined in the frequency domain.

- **Case 15904**

In version 12 the CLIP function has been modified so that when a curve is clipped new points are now inserted at the clip value by interpolating between the existing curve points. Previously any points with x-axis values outside the clipping region were thrown away and any points with y-axis values outside the region had the y-axis values adjusted to the clip region boundaries.

- **Case 5417**

The CLIP function now has options to pick the x and y min/max values interactively by clicking on the screen. Each value can be picked individually or an area defining the limits can be dragged out.

When picking points there is also an option to select the nearest curve point to where you click instead the actual screen coordinate.

Preferences

- **Case 25632**

T/HIS now has a '-pref' command line option that can be used to select a user defined preference file that is read after any other preference files.

```
-pref=filename
```

If 'filename' contains a PATH then it should be the complete path to the file, if it doesn't contain a PATH then it is assumed that the file is located in the installation directory.

Properties

- **Case 25061**

In version 11 it was possible to change the colour used to display injury criteria values (HIC, PHD etc) but it wasn't obvious how to do this (it is controlled by the Properties->Format menu).

In version 12 there are now 'Properties' buttons next to the options where the display of injury criteria are turned on and off which are a shortcut to the Format menu to make it easy to access these settings.

The option to control the colour has also been split up so there are now 2 separate settings:

'Text Colour' controls the colour of the injury criteria values and labels.

'Injury Marker Colour' controls the colour of the lines drawn for each injury value.

By default these are both set to the current foreground colour.

2 new preferences have also been added that can be used to set the defaults for these 2 options.

```
this*injury_line_colour:
this*injury_text_colour:
```

Quick Pick

- **Case 22255**

The Quick Pick menu in version 12 of T/HIS has 2 new options, 'Copy Curve(s)' and 'Paste Curves'. The COPY option can be used to select curves. Once curves have been selected with COPY the PASTE option can then be used to make a copy of the curves and to unblank them in just the graph that PASTE was selected in.

If for example you used COPY in graph 1 and dragged across curves 4 and 5 and then selected PASTE in graph 2 then 2 new curves would be created which were exact copies of 4 and 5 and would be unblanked in just graph 2.

- **Case 19019**

The Quick-Pick Operate functions popup now contains the Clip function. The CLIP function entry also has a new popup submenu that displays the x and y clipping values and can be used to change those values.

Read

- **Case 25506**

In version 12 of T/HIS models can be opened on a PC by using drag and drop. If files are dragged and dropped into the main application window T/HIS will check if the file is one of the LS-DYNA output files it reads and if it is then it will open all of the model files that it can find that go with the file. T/HIS also assumes that any files with a .cur file extension are T/HIS curve

files and tries to read them. Any files with a .csv file extension are treated as CSV files and the CSV read menu is automatically displayed.

- **Case 25496**

Recent files that have been read within T/HIS are now available via a dropdown on the file read input text box.

- **Case 25462**

Version 12 of T/HIS now has separate data components for reading the master and slave side contact forces. Previously T/HIS just has buttons for X,Y,Z and Magnitude and would always report the master side forces for contacts. Previously the master side was selected as in old versions of LS-DYNA node to surface contacts only reported forces on the master side.

Recent versions of LS-DYNA can generate forces on the slave side of some contacts (FORCE_TRANSDUCERS) while the master side reports zero force. Because of this T/HIS now has separate options for master and slave forces.

- **Case 25009**

The CSV file reader in T/HIS has been modified so that the field separator can now be changed from the default comma to either a tab or spaces.

If the 'space' option is used then multiple spaces are counted as a single field separator. If curve or axis labels are defined in the file and they contain spaces then they need to be enclosed in pairs of " quotes.

A new FAST-TCF command has been added to set the separator used when reading in a CSV file

```
define csv_separator tab
```
(options are comma, space or tab)

A new preference has also been added that can be used to change the default separator from 'comma' to either 'space' or 'tab'

```
this*csv_separator: Tab
```

- **Case 24634**

T/HIS 12.1 supports the 'Eroded Hourglass Energy' data component that is written to the GLSTAT ASCII file and LSDA (binout) file by some versions of LS-DYNA

- **Case 24523**

T/HIS 12.1 now supports the following INTOUT and NODOUT options on *DATABASE_BINARY_EXTENT

INTOUT : STRESS

This option generates stress tensor results for all integration points. For fully integrated shells this will generate results for all 4 in-plane integration points at each through thickness point. For fully integrated solids data for 8 integration points will be generated.

NODOUT : STRESS

This option will output stress tensors at the nodal points. These values are generated by extrapolating results from the integration points.

- **Case 23338**

T/HIS now supports the NODOUT option on *DATABASE_EXTENT_BINARY for writing element data values that have been extrapolated from the integration points to the nodes. This option applied to Solids, Shells and Thick Shells.

Currently only the STRESS, STRAIN and ALL options are supported.

- **Case 22139**

In Version 11 if you had associated '.cur' files with T/HIS then double clicking on a '.cur' file on a PC would start T/HIS and read the curve in. If you then wanted to read in another curve file you had to change the READ menu to CURVE before you could select the 2nd curve file.

From version 12 onwards the READ menu will automatically select the CURVE option if a curve file is read in when T/HIS starts by double clicking on the curve file.

- **Case 20557**

In some recent versions of LS-DYNA the shell and thickshell results in the LSDA (binout) file can be written using the Global coordinate system instead of the default local coordinate system (see EOCS on *CONTROL_OUTPUT). If the LSDA file also contains information for fully integrated shells and thickshells with multiple in-plane integration points then the file contains both ELOUT and ELOUTDET but the ELOUTDET results ignore the EOCS option and are still written using the local coordinate system.

If T/HIS detects both ELOUT and ELOUTDET in the LSDA file and they are using different coordinate systems T/HIS now displays a warning when the model is opened. By default when reading data T/HIS uses the ELOUTDET file if both are present as normally the data is ELOUT is a subset of that in ELOUTDET. If T/HIS detects a mix of coordinate systems it now offers the user a choice to ignore the ELOUTDET file when reading in Shell and Thickshell data so that the Global values in ELOUT can be read.

- **Case 12809**

If a user accidentally clicked on the 'Read > Keyboard' option then it wasn't obvious how to get out of the window that appears into which the curve information is entered.

Version 12 now has a 'Quit' button which closes the window without storing any curve data that has been added and a 'Finish' option that becomes active as soon as the 1st data point has been entered and saves the curve to the 1st free curve.

- **Case 11012**

If LS-DYNA incorrectly split the output for a single state across multiple THF files then T/HIS could fail to read the files correctly.

Scripting

- **Cases 23511**

A Datum class has been added to the T/His JavaScript API.

User Interface

- **Cases 18077 and 23772**

In version 12 the maximum length for filenames (including the directory) has been increased from 256 to 512 characters

NOTE : Internally Windows supports filenames of up to 32768 but currently it does not support pathnames longer than 260 characters within Windows Explorer or the command line interface.

3.3.2 Enhancements in 12.1

Menus

- **Case 27126**

If you read a model into T/HIS which has a groups file and you then read a 2nd model into T/HIS and it has a groups file then a window is displayed giving you a number of options.

This window makes it difficult to automate things using FAST-TCF and Reporter as you have to click in the window to select one of the options.

From version 12.1 onwards this menu will not be displayed if the environment variable 'MENU_AUTO_CONFIRM' has been set to 'true'.

Preferences

- **Case 27880**

In version 12.1 of T/HIS the default data sources for PART data has been changed from

THF > LSDA > ASCII

to

LSDA > ASCII > THF

This change has been made as there were some situations where T/HIS would be unable to correctly determine the location of the PART data in the THF file and could read and display incorrect data.

3.4 REPORTER

3.4.1 Enhancements in 12.0

Capture

- **Case 21309**

REPORTER will now prompt the user to replace variables in the macro after doing a capture in PRIMER. For each matched text string you can choose whether to replace it with a variable or you can do 'Yes to All' or 'No to All'

Scripting

- **Case 23990**

A new Batch() method has been added to JavaScript so scripts can test if REPORTER is running in batch mode

- **Case 22974**

New Item and Page classes have been added to JavaScript.

- **Case 17602**

A Duplicate method has been added to the Page class in JavaScript

- **Case 15772**

A System command has been added to JavaScript

User interface

- **Case 23503**

Objects can now be locked to prevent them from being moved on the page

3.4.2 Enhancements in 12.1

There have not been any enhancements in 12.1